

# **MAINVIEW® for IMS Online Analyzers Reference Manual**

**Version 3.3**

**November 2002**



Copyright 1998–2002 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. IBM is a registered trademark of International Business Machines Corp. All other registered trademarks or trademarks belong to their respective companies.

## Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

---

## Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

### United States and Canada

**Address** BMC Software, Inc.  
2101 CityWest Blvd.  
Houston TX 77042-2827

**Telephone** 713 918 8800 or  
800 841 2031

**Fax** 713 918 8000

### Outside United States and Canada

**Telephone** (01) 713 918 8800

**Fax** (01) 713 918 8000

---

---

## Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

### Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at <http://www.bmc.com/support.html>. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

### Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at <http://www.bmc.com/support.html>.

### Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software



---

# Contents

How to Use This Book .....	xv
----------------------------	----

---

## Part 1. Performance Analysis and Monitoring Techniques..... 1

<b>Chapter 1. Optimizing System Workflow .....</b>	<b>3</b>
MVIMS Usage Modes .....	3
Message Format Service (MFS).....	4
Analyzing MFS .....	4
Monitoring MFS .....	5
Suggestions .....	6
Queuing .....	6
Analyzing Queuing .....	6
Monitoring Queuing .....	8
Scheduling .....	8
Analyzing Scheduling .....	8
Class Queuing .....	8
Balancing Group (BALG) Queuing.....	9
Scheduling Activity .....	9
Monitoring Scheduling .....	10
Pool Utilization .....	10
PSB and DMB Pools .....	10
Pool Utilization for LSO=S .....	11
Monitoring Pool Utilization.....	11
Suggestions .....	11
Analyzing Operational Displays.....	12
Total System Monitoring.....	12
Dependent Region Monitoring .....	14
Analyzing Databases .....	16
OSAM Buffer Pool .....	16
VSAM Buffer Pool .....	17
Hit Ratios .....	17
VSAM Hiperspace .....	18
Fast Path Buffer Pool.....	18
Points Common to All Buffer Pools .....	19
Suggestions:.....	19
Analyzing IMS Internals .....	21
Latches .....	21
Suggestions for Relieving Latch Conflicts:.....	21
Logs.....	21
Online Log Data Set (OLDS) .....	21
Write-ahead Data Set (WADS) .....	22
System Log Data Set (SLDS) .....	22
Log Performance Suggestions .....	22
Program Isolation Analysis.....	23
Pools .....	24
CBT Pools .....	24
Non-CBT Pools .....	24
Analyzing the System.....	27
Dispatching .....	27
Real Storage .....	28

<b>Chapter 2. User Techniques for Monitors</b> .....	29
Master Terminal Operator (MTO) .....	30
BLKMTO Example .....	30
IMS Manager .....	32
BLKMGR Example .....	32
IMS Performance Analyst and System Programmer .....	33
BLKPERF Example .....	33
Database Administrator .....	35
BLKDBA2 Example .....	35

---

## **Part 2. Using MVIMS** .....

<b>Chapter 3. The MAINVIEW Product Family</b> .....	39
<b>Chapter 4. Measuring IMS Activities and Resources</b> .....	41
<b>Chapter 5. Types of Services</b> .....	49
Requests .....	50
Analyzers .....	51
Monitors .....	52
Starting and Stopping Monitors .....	52
Monitor Response to Target IMS Shutdown and Startup .....	52
Data Collection .....	52
Historical Data Stored .....	53
Data Types Measured .....	54
Warning Conditions .....	55
Monitor Request Title .....	56
Warning Message Format .....	56
IMS Monitor Data Display Services .....	58
Workload Wait Events .....	59
Workload Trace .....	59
Logging a Display .....	60
Automatic BBI-SS PAS Image Logging of Analyzer Displays .....	60
Automatic BBI-SS PAS Image Logging of PLOT Display .....	60
Automatic BBI-SS PAS Image Logging of Monitor Summary Displays .....	61
Request Status .....	62
Grouping Requests .....	62

---

## **Part 3. Analyzers** .....

<b>Chapter 6. Displaying a List of Analyzer Services (Menu Option 1)</b> .....	67
SORT Primary Command .....	68
AREA Primary Command .....	69
Line Commands .....	69
Start Image Log Request (I Line Command) .....	71
<b>Chapter 7. Requesting an Analyzer Display</b> .....	73
Selecting a Service(s) for Display .....	73
Logging an Analyzer Service Display .....	74
Stopping an Analyzer Image Log Request .....	75
Qualifying an Analyzer Request .....	75

<b>Chapter 8. Resource Analyzer Services (Quick Reference)</b>	77
<b>Chapter 9. Message Format Service Displays</b>	83
MFSST - MFS Statistics	83
MFSUT - MFS Pool Utilization	87
<b>Chapter 10. Queuing Displays</b>	89
QUEST - Queue Statistics	89
<b>Chapter 11. Scheduling Displays</b>	95
BALGQ - BALG Queuing	95
CLASQ - Class Queuing	102
PSBUT, DMBUT - PSB and DMB Pool Utilization	109
SCHED - Scheduling Statistics	112
<b>Chapter 12. Region Displays</b>	115
DLIST - DL/I Call Status	118
DREGN - Region Detail (Event Collector Data)	125
DREGN - Region Detail (No Event Collector Data)	145
REGNS - IMS Regions (Event Collector Data)	164
REGNS - IMS Regions (No Event Collector Data)	187
STAT/STATR - System Status	206
<b>Chapter 13. Database Displays</b>	215
DBST - OSAM Global Pool Statistics	215
DBST - OSAM Subpool Statistics	219
FPBST - Fast Path Buffer Pool Statistics	223
VSST - VSAM Global Pool Statistics	226
VSST - VSAM Subpool Statistics	229
<b>Chapter 14. IMS Internals Displays</b>	233
APPCA - APPC Activity Summary	233
APPCL - APPC LU Status	242
DAPPC - Input Allocation Direction	249
DAPPC - Output Allocation Direction	254
DLTCH - Latch Detail	258
DPOOL - Detail Pool (Non-CBT Fixed Pool)	262
DPOOL - Detail Pool (Non-CBT Variable Pool)	268
LATCH - Latch Summary	272
LOGST - Log Statistics	279
PI - Program Isolation	284
POOLC - Pool Summary (CBT)	288
POOLS - Pool Summary (Non-CBT Variable and Fixed Pools)	291
<b>Chapter 15. OS/390 System Displays</b>	299
DSPST - Dispatcher Statistics	299
RS - Real Storage	303
<b>Chapter 16. IRLM Displays</b>	307
IRLM - IRLM IMS Status (IRLM 1.5)	307
IRLMG - IRLM GLOBAL STATUS (IRLM 1.5)	313
LCRES - IRLM Lock Contention by Resource (IRLM 1.5)	321
LCUSR - IRLM Lock Contention by User (IRLM 1.5)	328
LHRES - IRLM Locks Held by Resources (IRLM 1.5)	330
LHUSR - IRLM Locks Held by User (IRLM 1.5)	337

LUSRD - IRLM Lock User Detail (IRLM 1.5) .....	339
IRLM - IRLM IMS Status (IRLM 2.1 and Later) .....	347
IRLMG - IRLM GLOBAL STATUS (IRLM 2.1 and Later) .....	353
LCRES - IRLM Lock Contention by Resource (IRLM 2.1) .....	362
LCUSR - IRLM Lock Contention by User (IRLM 2.1) .....	370
LHRES - IRLM Locks Held by Resources (IRLM 2.1) .....	372
LHUSR - IRLM Locks Held by User (IRLM 2.1) .....	380
LUSRD - IRLM Lock User Detail (IRLM 2.1) .....	383
 <b>Chapter 17. WORKLOAD ANALYZER Displays (Quick Reference)</b> .....	 393
 <b>Chapter 18. ISTAT - Terminal Input Status Display</b> .....	 395
Input LTERM Status .....	398
 <b>Chapter 19. OSTAT - Terminal Output Status Display</b> .....	 407
Output LTERM Status .....	412
 <b>Chapter 20. TRANQ - Transaction Queue Status Display</b> .....	 419
Transaction Status .....	423
 <b>Chapter 21. USER - User Status Summary</b> .....	 427
 <b>Chapter 22. Requesting Workload Wait Data Collection (MWAIT)</b> .....	 433
Starting MWAIT .....	433
Using the Workload Wait Data Entry Panel .....	435
Stopping MWAIT .....	440
Storage Requirements .....	440
Defining a Sampling Interval .....	441
Qualifying a Workload MWAIT Request .....	441
 <b>Chapter 23. Requesting Workload Wait Data Display (DWAIT)</b> .....	 443
 <b>Chapter 24. DWAIT - Workload Wait Display</b> .....	 445
Excluding Queued Events .....	450
DWAIT Event Definitions .....	450
Application Program (AP) Events .....	451
Input Communications (IC) Events .....	468
Input Queue (IQ) Events .....	477
Output Communications (OC) Events .....	480
Output Queue (OQ) Events .....	481
Scheduling (SC) Events .....	484
Synchronization Point (SP) Events .....	498

---

<b>Part 4. Appendixes</b> .....	509
 <b>Appendix A. IMS Dump Analysis</b> .....	 511
AO Exit .....	511
MAINVIEW AutoOPERATOR for IMS Routines in IMS .....	511
Event Collector .....	511
 <b>Appendix B. How Product Libraries Should Be Used</b> .....	 513
 <b>Index</b> .....	 515

---

# Figures

- 1. List Analyzer Display Services Application .....67
- 2. Image Log Request Application ..... 71
- 3. Start Workload Wait Request. ....435



---

# Tables

- 1. Services Supporting Hit Ratios ..... 17
- 2. IMS Activities and Resources Measured. .... 42
- 3. Resource Analyzer Service Select Codes ..... 77
- 4. Workload Analyzer Display Service Select Codes ..... 393
- 5. SET Keywords to Activate a Workload Wait (MWAIT) Service ..... 436
- 6. Latch Locks ..... 461
- 7. Latch Locks ..... 469
- 8. Latch Locks ..... 489
- 9. Product Libraries ..... 513



---

## How to Use This Book

This manual documents the features and functions of the MAINVIEW® for IMS (MVIMS) Online Resource Analyzer and Workload Analyzer display services.

The features, and functions of the MVIMS Online Resource Monitor, Workload Monitor, and trace services are documented in the *MAINVIEW for IMS Online – Monitors and Traces Reference Manual*.

The features, and functions of the IMSplex System Manager (IPSM) component are documented in the *MAINVIEW for IMS Online – IPSM Reference Manual*.

For information about what's new in the current release of MAINVIEW for IMS Online, see the product Release Notes, which are available on the BMC Software Support Web pages.

This manual is intended for use by the IMS master terminal operator (MTO), system programmer, database administrator, or performance analyst who monitors the status, activity, and performance of IMS and its resources.

---

## MVIMS Product Library

MVIMS is integrated with the BMC Software MAINVIEW® architecture. MAINVIEW is a base architecture that allows authorized users to use a single terminal to interrogate any OS/390, CICS, IMS, DB2, or MQSeries subsystem in a sysplex.

The MVIMS product library includes the following documents.

### MAINVIEW for IMS **Online:**

*MAINVIEW for IMS Online – Customization Guide*

*MAINVIEW for IMS Online – IPSM Reference Manual*

*MAINVIEW for IMS Online – Analyzers Reference Manual*

*MAINVIEW for IMS Online – Monitors and Traces Reference Manual*

*MAINVIEW for IMS Online – Release Notes*

### MAINVIEW for IMS **Offline:**

*MAINVIEW for IMS Offline – Customization and Utilities Guide*

*MAINVIEW for IMS Offline – Performance Reporter Reference Manual*

*MAINVIEW for IMS Offline – Transaction Accountant Reference Manual*

*MAINVIEW for IMS Offline – Release Notes*

---

## How This Book Is Organized

This manual is divided into the following parts:

- Part 1 describes techniques to help you use MVIMS to optimize IMS performance. It contains references to specific services.
- Part 2 describes how to use MVIMS. It includes descriptions of
  - Analyzers and monitors and how they are used
  - The online functions that can be used with MVIMS in a terminal session (TS)
  - The Primary Option Menu and applications that provide easy access to product service applications
- Part 3 describes how to use each of the analyzer display services.
- Part 4 has appendixes that provide information about how to analyze IMS dumps and how to use the MAINVIEW distributed product libraries.

To help you find information about a service quickly:

- Service descriptions are arranged alphabetically by name and grouped by sections that correspond to a major IMS transaction processing event.
- Service parameters, display fields, and DWAIT display events are described in alphabetical order.
- All of the service names are in alphabetical order together as indexed entries to “Service Select Code” in the index.

---

## Related MAINVIEW Products

The related MAINVIEW-based products include the following:

- MAINVIEW® AutoOPERATOR™
- MAINVIEW® for CICS
- MAINVIEW® for DB2®
- MAINVIEW® for DBCTL
- MAINVIEW® FOCAL POINT
- MAINVIEW® for MQSeries
- MAINVIEW® for OS/390
- MAINVIEW® VistaPoint™

Customization and administration instructions for the MAINVIEW-based functions are provided in the *MAINVIEW Common Customization Guide*. The following manuals document product-specific customization instructions:

- *MAINVIEW AutoOPERATOR Customization Guide*
- *MAINVIEW for CICS Customization Guide*
- *MAINVIEW for DB2 Customization Guide*
- *MAINVIEW for DBCTL Customization Guide*
- *MAINVIEW for IMS Online – Customization Guide*

- *MAINVIEW for IMS Offline – Customization and Utilities Guide*
- *MAINVIEW for OS/390 Customization Guide*

The following books document the use of general services common to MAINVIEW for IMS and related products:

- *MAINVIEW AutoOPERATOR Basic Automation Guide*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*
- *MAINVIEW for CICS PERFORMANCE REPORTER User Guide*
- *MAINVIEW for DB2 User Guide* (Volumes 1, 2, and 3)
- *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*

---

## Related IBM Publications

*OS/390 Initialization and Tuning Guide*  
*IMS Operator Reference*  
*System Administration Guide*

---

## Conventions Used in This Manual

The following symbols are used to define command syntax, are *not* part of the command, and should never be typed as part of the command:

- Brackets [ ] enclose optional parameters or keywords.
- Braces { } enclose a list of parameters; one must be chosen.
- A line | separates alternative options; one can be chosen.
- An underlined parameter is the default.

The following command syntax conventions apply:

- An ITEM IN CAPITAL LETTERS must be typed exactly as shown.
- Items in *italicized, lowercase* letters are values that you supply.
- When a command is shown in uppercase and lowercase letters, such as **HSplit**, the uppercase letters show the command abbreviation that you can use (**HS**, for example). The lowercase letters complete the entire command name. Typing the entire command name is an optional, alternative way of entering the command.
- Commands without an abbreviation (**END**, for example) appear in all uppercase letters.



---

## Part 1. Performance Analysis and Monitoring Techniques

This section describes techniques to help you use MVIMS to optimize system performance.

<b>Chapter 1. Optimizing System Workflow</b>	3
MVIMS Usage Modes	3
Message Format Service (MFS)	4
Analyzing MFS	4
Monitoring MFS	5
Suggestions	6
Queuing	6
Analyzing Queuing	6
Monitoring Queuing	8
Scheduling	8
Analyzing Scheduling	8
Class Queuing	8
Balancing Group (BALG) Queuing	9
Scheduling Activity	9
Monitoring Scheduling	10
Pool Utilization	10
PSB and DMB Pools	10
Pool Utilization for LSO=S	11
Monitoring Pool Utilization	11
Suggestions	11
Analyzing Operational Displays	12
Total System Monitoring	12
Dependent Region Monitoring	14
Analyzing Databases	16
OSAM Buffer Pool	16
VSAM Buffer Pool	17
Hit Ratios	17
VSAM Hiperspace	18
Fast Path Buffer Pool	18
Points Common to All Buffer Pools	19
Suggestions:	19
Analyzing IMS Internals	21
Latches	21
Suggestions for Relieving Latch Conflicts:	21
Logs	21
Online Log Data Set (OLDS)	21
Write-ahead Data Set (WADS)	22
System Log Data Set (SLDS)	22
Log Performance Suggestions	22
Program Isolation Analysis	23
Pools	24
CBT Pools	24
Non-CBT Pools	24
Analyzing the System	27
Dispatching	27
Real Storage	28

<b>Chapter 2. User Techniques for Monitors</b> .....	29
Master Terminal Operator (MTO).....	30
BLKMTO Example.....	30
IMS Manager .....	32
BLKMGR Example .....	32
IMS Performance Analyst and System Programmer .....	33
BLKPERF Example .....	33
Database Administrator .....	35
BLKDBA2 Example .....	35

---

# Chapter 1. Optimizing System Workflow

This chapter describes how MVIMS can be used to analyze and monitor system operation and performance. The chapter is organized into sections by IMS processing event components: MFS, queuing, scheduling, operations (region activity), database, IMS internal functions, and OS/390 functions. Within each section, a performance problem is described, along with the applicable analyzer or monitor solution.

---

## MVIMS Usage Modes

MVIMS can be used in two different modes:

- For ongoing analysis of what is currently happening in IMS

This mode provides a realtime picture of the current state of the system. These realtime displays are provided by Resource Analyzer and Workload Analyzer services.

Analyzer services can be time-driven. In this mode, you can refresh the displays in a time-driven cycle or log them automatically at time-driven intervals to a BBI-SS PAS Image log for later retrieval.

- As a monitor to assist you in tuning the performance of the whole system by investigating the functions and resources over time that may be bottlenecks

This mode provides statistics accumulated at user-specified time intervals. Time-driven IMS workload samplings and graphic plot displays of the collected data samplings are provided by Workload Monitor and Resource Monitor services. IMS workload wait event and transaction trace data are collected and displayed by Workload Analyzer.

Monitor-collected values can be compared to user-specified thresholds and warning messages can be generated. A warning message can invoke automatic corrective action from MAINVIEW AutoOPERATOR or alert a user to take corrective action.

You can set MVIMS services to monitor IMS performance at different times of the day or on different days of the week when processing characteristics may change. Small samples can be taken over time and comparisons made to determine the best performance indicators. Service sampling of IMS performance can help you determine if some changes are caused by application program or system design errors or oversight. Concentration should be directed in the areas where changes have the most effect in system performance.

Often it is valuable to know what has happened in a particular time interval, such as the last five to ten minutes. You can determine time interval information by making two observations and calculating how many events occurred between the two. By relating these numbers to the length of the time interval chosen, you can calculate rates, such as requests-per-second or I/O-per-second. You can calculate rates such as I/O-per-request by comparing the number of events to any other counter (which is also possible for a specific time span). Such calculations require more work but result in more meaningful measurements and numbers that can be compared over time to show changes in the system.

With Resource Monitor, calculations are performed automatically for the most important system variables, and the measurement made by each calculation can be displayed with the PLOT service. If several monitor requests are set up for the same sampling interval, the current measurement values can be viewed and compared easily with the DMON service.

Some subsections in this chapter include suggestions for ways to improve performance. The suggestions should be used as guidelines, and shouldn't be interpreted as the best, or only, actions to be taken. A suggestion may not be valid in every situation and must be considered for its value in the particular installation and mode of operation.

---

## Message Format Service (MFS)

This section describes analyzing and monitoring MFS.

### Analyzing MFS

Message Format Service (MFS) is the first major IMS function encountered by an incoming request and the last function encountered for an outgoing response. MFS can have a great impact on the efficiency and productivity of the entire system. As one of the unique features of IMS, the generality and flexibility of MFS processing can result in performance problems if misused. For optimum performance, define only as many MFS formats as will fit in the MFS pool.

The formats reside online as records in a PDS with RECFM=U. The MFS data set should be allocated as a single extent by cylinders to an area without alternate tracks. The volume should be mounted private on nonshared DASD.

When a format is requested that is not in the pool, MFS must locate the directory entry for the format. If there is an entry for it in the in-core directory `$$IMSDIR`, a directory read is not necessary to retrieve the entry from the directory block.

The directory entry contains the TTR of the first record of the format and a half word of user data that contains the total length of the format block. Using this length, space is obtained in the MFS pool to contain the block. The format is then read into the obtained area, one record at a time. The format block may be split into multiple records if the block size is too small. The block size should be at least as large as the largest format block because additional I/O is required to retrieve these multi-record formats.

The formats are maintained in the pool as long as possible to reduce the amount of I/O. Only when space is required in the pool does MFS free the least-recently-referenced unused format block. This release continues until sufficient contiguous free space has been obtained to satisfy the request. Fragmentation can be a problem in any pool of variable length blocks managed in this manner. The Message Format Pool Utilization service, MFSUT, can be used to study the effect of space release upon the MFS pool, including fragmentation. For more information about MFSUT, see the service description on page 87.

The basic request types that can be made to MFS are

- PRE-FETCH REQUEST(S)
- IMMEDIATE REQUEST(S)
- FREE BLOCK REQUEST(S)

PRE-FETCH is an anticipated future need for a format block. If the pre-fetch feature is enabled, it informs PRE-FETCH to retrieve the requested format block.

An immediate request is for a block that must be read into the MFS buffer pool before processing can continue. The IMMEDIATE REQUEST(S) counter is a good indicator of MFS activity, especially in relation to total MFS reads, and the counter is useful for evaluating MFS performance. It is possible to calculate the average physical I/O-per-second to the MFS data set. Depending on the device and contention, this statistic may or may not indicate a problem.

FREE BLOCK REQUEST(S) inform the MFS pool handler that the format block is no longer being actively referenced and is now a candidate for being washed from the pool if space is needed for another block.

The MFSST service (which is described on page 83) displays these counters in matrix format. Each line represents one of the three request types:

- PRE-FETCH REQUEST
- IMMEDIATE REQUEST
- FREE BLOCK REQUEST

Two columns represent the major queues where the FRE could have been located:

- IMMEDIATE QUEUE
- FREE BLOCK QUEUE

Currently, the counter is incremented when the FRE is located in a queue, which does not necessarily mean that the format block itself is already in the pool. The MFS fetch request handler DFSFFRH0 could be modified to check for a loaded format block before incrementing the counter, which could lead to a better indication of MFS performance. The percent of IMMEDIATE REQUESTs satisfied in the pool (in the IMMEDIATE or FREE BLOCK queues) is a good indicator of how well MFS is performing. The larger the percentage, the fewer synchronous waits for format I/O.

The MFS pool is above the 16Mb line. Because of virtual storage constraint relief (VSCR) above the 16Mb line, the MFS pool can be increased by 400 to 500 percent to further reduce the number of I/Os to the FORMAT data set. If the pool size is increased, be sure to monitor the number of FREs because it may be necessary to increase them to utilize the additional space in the pool. The directory entries are built dynamically in the pool as they are used. The \$SIMSDIR is still used, but is not as significant as in previous releases. These dynamic entries are flushed whenever an online change is made to the MFS formats.

## Monitoring MFS

MFS performance can be monitored continuously by these Resource Monitor monitors:

MFSIO	MFS input/output requests
MFSIR	MFS immediate requests
MFSFD	Percent of MFS blocks found in pool

If any of these values are considered to be excessive in given time periods, additional investigation is required (see “Suggestions” on page 6). There are no specific values that are good or bad; each system must be evaluated individually.

## Suggestions

- Optimize the pool space and FRE allocation as described.
- Use the in-core format directory `$$IMSDIR`.
- Eliminate any unnecessary or unused format blocks.
- After updating/deleting a format, compress the library.
- Allocate by cylinder (but no more space than necessary) and make certain there are no alternate tracks.
- Watch the placement of the MFS data set and mount private on nonshared DASD.
- Make certain that the block size is at least as large as the largest format.
- Do not allocate any more directory blocks than necessary.
- If the MFS data set is large, consider reordering the formats. You can determine an optimal order with one of these methods:
  - Analyze the IMS log to determine the frequency of use.
  - Analyze the DC Monitor output to calculate the frequency fetched.

You can then build the MFS data set by generating the formats in order of decreasing use. Use the frequency fetched from the DC Monitor to reorder the individual format blocks.

---

## Queuing

This section discusses analyzing and monitoring queuing.

### Analyzing Queuing

The next major IMS function that can be a major system bottleneck is the queuing of input requests and output responses. Like MFS, the queuing routines of IMS attempt to keep as much as possible in the queue pool to reduce I/O. I/O is done only when checkpoints request a pool purge, when space is needed in the pool, or when something previously written out must be retrieved. Thus, the queue pool and data sets can be critical to good performance.

Queuing makes use of preformatted OSAM data sets.

- Each data set should be allocated by cylinder as a single extent on an area without any alternate tracks.
- These data sets should be mounted private on a low-contention, nonshared device.

The optimization of the queue pool parameters is a more difficult problem. Several problem areas are:

- There are constraints that limit the possible values that can be defined. IMS imposes a strict relationship between the block size (BLKSIZE) and logical record length (LRECL) of the three queue data sets (space is wasted unless the block size is a multiple of all three LRECLs).

- The minimum LRECL of the LONG message queue data set frequently is dictated by the applications. The length of segments being queued to and from the applications determines the optimal choice of the three parameters.
- The LRECL for the SHORT message queue data set is especially difficult to determine. If this LRECL is too large, the SHORT message queue data set will be overutilized. If the LRECL is too small, the LONG message queue data set will be overutilized. Both situations waste space.

MVIMS can point to a possible problem in this area, but the solution may require offline analysis to determine the average segment length or the segment length distribution. These online Analyzer displays are helpful:

- Service DREGN (PSB/Transaction area of display) shows the average length of all input messages of the transaction types currently processing (as calculated by the IMS queue manager).
- Service STAT shows the utilization of each of the three queue data sets. By observing these percentages for a period of time, an imbalance between SHORT and LONG can be detected easily.

To maximize the use of the queue pool and reduce I/O, messages that remain queued for a long time should be set up as candidates to wash out. This can be done by setting the record length of the LONG message queue data set equal to the common block size and defining the segment length of such messages long enough to force them into the LONG message queue. This allows a block filled by such a message to be written out immediately, releasing the block. Otherwise, the segment takes up only one record in the block and the block may remain in the pool, but only the remaining records can be used.

The types of messages that should be considered are transactions that normally do not schedule (locked, stopped, or priority zero), and long output messages that are routed to slow remote printers or that use terminal operator paging. Any space wasted on the data set by possible padding is of minor importance in comparison to the better use of the queue buffers in the pool.

The remaining parameter is the determination of the number of queue buffers. Insufficient queue buffers results in unnecessary queue manager I/O. Excessive queue buffers wastes central storage and increases the paging rate on a real-storage constrained system.

Although the TOTAL REQUESTS to the queue manager is the best indication of the amount of queuing activity, the number of ENQUEUEES plus CANCELS gives a better idea of the number of messages involved. These messages can be either single/multiple segment input requests or output responses. Using this sum, the amount of queuing activity per message can be calculated (TOTAL REQUESTS/(ENQUEUEES + CANCELS)).

It is also possible to calculate the amount of queue manager activity per second. REPOSITIONS is a nonproductive call to the queue manager used to re-examine a previous segment. The number of PCBs UNCHAINED is incremented for each PCB that loses position when a buffer is written out. Finally, the number of WAITS FOR AN AVAILABLE BUFFER is incremented when a request had a wait while a buffer is written out to release space. As with many statistics, they are more meaningful when viewed as events per second or events per unit of work (such as I/O per message).

## Monitoring Queuing

There are 15 Resource Monitor services for monitoring queuing data (see Part 3, “Monitors,” in the *MAINVIEW for IMS Online – Monitors and Traces Reference Manual*). Workload Monitor services monitor the input queue time by various selection criteria, such as class, transaction code, LTERM, region, program, and USERID. With well-considered threshold specification, these monitors produce warning messages that can be used to trigger automatic operational changes or alert operations personnel to use analyzer services to investigate further.

---

## Scheduling

This section describes analyzing and monitoring scheduler pool utilization.

## Analyzing Scheduling

The scheduling function of IMS (selecting and preparing application programs to run in each dependent region as it completes its previous task) is very complex and critical for good performance of the system. It is dependent on several factors that you can directly influence (for example, pool sizes and transaction class assignments). However, the basis of information for making decisions that have positive results on this process are difficult to obtain when it is needed. MVIMS services such as SCHED, STAT, and DSPST address several of the major problem areas.

## Class Queuing

IMS queues and schedules transactions according to class assignments and by priority within class if needed. Each transaction is defined to the system as belonging to one class. Each message region has from one to four specified classes that it can process. This allows you to balance the processing load, give priority to transactions with critical response time requirements, isolate long-running transactions, and so on.

The CLASQ display presents an overview of the current situation: what is queued in each of the classes, what the regions are doing currently, and what classes each region can accept.

By examining the status of the queues at various times of the day, any imbalance caused by the class assignments should be indicated by excessive queue size for some classes or by idle regions waiting for input in other classes. If such imbalances occur often (perhaps only at certain times of the day), performance can be improved either by reassigning transaction classes or by dynamically changing the region processing classes. Because this display shows the status of all the classes at once, the effect of such reassignments can be detected easily and followed through time.

**Note:** An increasing queue delay caused by load imbalance (check the enqueue time (ENQ TIM) of the current transactions being processed) also impacts the system by increasing the overhead incurred by the queue manager. The longer the queues, the greater the possibility that a transaction will be washed out of the pool to the queue data sets before it is needed, causing extra I/O activity to write it out and to retrieve it. The QUEST service can be used to investigate this occurrence (see “Analyzing Queuing” on page 6). See the information in the REPOSITIONS or PCBs UNCHAINED fields of the QUEST display.

## Balancing Group (BALG) Queuing

IMS Fast Path transactions do not use the message queues. Instead of the message queuing mechanism, each physical terminal has an expedited message handler (EMH) buffer associated with it. When an input message is received by IMS and is defined either as Fast Path exclusive or Fast Path potential, control is given to a user exit that can modify the routing code. Assign a different routing code or indicate that the message should be processed as a full function IMS transaction; that is, use normal message queuing.

Once a Fast Path routing code is assigned to an input message, the balancing group (BALG) is determined and the EMH buffer is queued to the BALG for processing.

This queuing takes place only if the BALG is active. The EMH buffers are processed in a FIFO sequence by the Fast Path regions that service the BALG. More than one region can process the same BALG. By observing the data displayed by the BALGQ service, it is possible to determine when there is a delay in processing for a specific BALG by the presence of a queue. This problem can be remedied by increasing the number of regions processing the BALG.

## Scheduling Activity

When a region is free and input belonging to one of its classes is available, IMS attempts to schedule a transaction. Many checks are necessary to ensure successful scheduling, and if any of these checks fail, that region may have to wait for some event before it can become active. However, in this case, the attempt is made first to schedule a different transaction.

Monitoring the success of system scheduling over time (by viewing the percentages of schedules and failures on the Resource Analyzer SCHED display or calculating the number of schedulings per minute) can point out possible problems before they become critical or make reduction of hidden overhead possible.

- **PROGRAM CONFLICTS** can be reduced by allowing parallel processing (load balancing); however, this can cause PI conflicts because of database intent. Care must be taken to minimize both of these items.
- If **PRIORITY CUTOFFS** are occurring regularly, check these definitions carefully to see if it is actually necessary to bypass available transactions (this can cause a region to be idle) to wait for the higher priority transaction.
- **INTENT FAILURES** can be caused either by insufficient space in the PSB, PSB work, or DMB pools, or by database intent failures. Excessive intent failures can indicate that too many regions are competing for the available resources, increasing the time needed for each, both in scheduling and in processing. This could be the result of allowing parallel scheduling (see “Program Conflicts” above).

**Note:** Currently, database intent failures occur only when a PROCOPT of EXCLUSIVE is used.

- **OTHER REASONS** for failures include locked or stopped transactions, programs, or databases.

**Note:** The associated SMBs are only removed from scheduling queues if the program is bad (BLDL failed) or a checkpoint purge is in progress. Although locked transactions no longer appear as schedulable in the CLASQ display, the SMBs are still examined and this counter is incremented. To prevent this counter from being continuously incremented, assign such transactions to an unassigned processing class.

### **Suggestions**

- Complicated class and priority assignments or the use of many dependent regions increases the contention during scheduling. The simple way is often the best.
- Use the processing limit for key transactions to reduce the number of schedulings. The same program being rescheduled causes a program fetch each time (unless preloaded), and is very expensive.
- Consider wait-for-input (WFI) processing for frequently used transactions. This dedicates a dependent region(s) to a transaction and eliminates program fetch.

## **Monitoring Scheduling**

Several Resource Monitor monitors are available to monitor arrival rates and processing that can affect scheduling. For example, it might be necessary to stop a less important transaction or BMP if the arrival rate of a particular transaction exceeds a user-defined threshold. This can be done either by an MAINVIEW AutoOPERATOR for IMS EXEC or through manual intervention.

## **Pool Utilization**

This section describes PSB and DMB pools and pool utilization for LSO=S.

### **PSB and DMB Pools**

The two main pools concerned in scheduling contain the IMS control blocks that define the application program's logical databases (PSBs) and the physical databases they access (DMBs) for the DL/I (IMS Data Language/I) interface. If not already present, the necessary blocks must be loaded into these pools during scheduling. A pool space failure can cause a region being scheduled to wait until resources are released by the completion of work in another region. Check the intent failure counter in the SCHED display.

If a DMB must be washed from the pool to free space for another, the associated data sets must be closed (a very time-consuming process that should be avoided). If that DMB is ever accessed again, it is necessary to reload it and open the associated data sets. If LSO=Y is coded, all open/close activity occurs in the IMS control region.

The PSBUT and DMBUT displays (see the service descriptions on page 109) are designed to show the free space still available in the pools and fragmentation and its potential impact. The displays show the current status of the pool and simulate the results of applying the least-recently-referenced space release algorithm to the blocks not currently being used. Each line displays successively the number of allocated blocks and the number of free spaces in the pool with minimum, average, maximum, and total lengths.

With these displays, it is possible to see not only the current free space (first line), but also the total free space available if needed (last line). This free space would be gained by freeing allocated but unused blocks. The maximum free space column shows the largest block that could be loaded into the pool (compare this to the maximum defined block size).

## Pool Utilization for LSO=S

Open/close activity takes place in the DLISAS region and is still DLISAS time-consuming, but it does not impact IMS work in the control region. If LSO=S is selected, the PSB pool is split into two separate pools. One of these pools resides in CSA and the other resides in the DL/I subordinate address space (DLISAS). The PSBUT service can display utilization of either pool.

The CSA pool contains the TP PCBs and the Fast Path PCBs. The DLISAS private-area pool contains the full function PCBs. This division of pools results in a split of approximately 20 percent in CSA and 80 percent in private storage.

## Monitoring Pool Utilization

The following Resource Monitor services can be used to monitor usage thresholds for these pools:

PIPL	Program Isolation
WKAP	General Work Area WKAP pool
DBWP	Database Work Area DBWP pool
PSBW	PSB Pool
DMBP	DMB Pool
MFSP	MFS Pool

If thresholds for these pools are consistently exceeded, consideration should be given to increasing the appropriate pool. Keep in mind that increasing pool sizes may strain another resource such as real storage (paging) or CSA.

## Suggestions

- Check the DMBUT display over time to see if the pool is remaining stable. If not, DMBs may be washing out of the pool and incurring large overhead. This can be investigated by using the IMS log tape utility to check for log record types 20 and 21 (database open/close).
- To reduce fragmentation and the possibility of a pool space failure, heavily used DMBs should be made resident. Heavily used PSBs and the intent lists should also be made resident. This causes them to be placed in separate areas, not in the pools.
- Use the DOPT option of the APPLCTN macro only on a test system. It results in a directory search of ACBLIB to locate the PSB every time such a program is scheduled.
- Access to the ACBLIB may affect scheduling and thus system performance. Many of the suggestions made in “Message Format Service (MFS)” on page 4 regarding the MFS format library are applicable.
- The following points should be considered to help reduce program fetch activity:
  - The default number of entries in the BLDL list (dynamic entry count) is 20. This can be increased at region startup with the parameter DBLDL=nnn to reduce directory searches. It should be set to zero on a test system to ensure that the newest copy is always fetched.

- Preloading is especially effective for small, frequently used subroutines (for example, COBOL and PL/I subroutines). It is not necessary for these modules to be link edited as re-entrant or reusable, but they must be so logically. Paging probably would adversely affect large application programs that are preloaded.
- The steplibs for each dependent region should be concatenated with the most frequently used application program libraries first and the IMS RESLIB last. The application program libraries should be full-track blocked to reduce reads.
- Consider using the OS/390 VLF feature to manage program loading.

---

## Analyzing Operational Displays

There are six standard operational displays of data concerning the functioning of the IMS dependent regions. The *MAINVIEW for IMS Online – Customization Guide* describes how to modify displays or include others tailored to site requirements. In addition, the system status display (STAT) provides an overview of the status of the total IMS.

These comprehensive region displays show the work that the IMS system is currently performing. They should be used to monitor this activity regularly so that problem areas can be located as they occur (for example, a program loop in a dependent region). These displays also show up to eight regions concurrently so that activity in different regions can be compared.

## Total System Monitoring

The system status display (STAT/STATR) is designed to simplify continuous monitoring of the system, pointing out possible problem areas that can be investigated in detail with the other displays, or confirming that all is running as smoothly as it should. If there are standard recognizable problems, warning indicators are set and highlighted. Other problem areas may be recognized only in light of knowledge about specific operating characteristics of a particular system. Therefore, the data displayed should be scanned even when no warnings are shown. If the asynchronous services option is available, many variables can be monitored automatically and warning messages sent when user-defined thresholds are exceeded.

There are two basic parts to this display:

- An overview of the critical resources that allow work to be accomplished
- An overview of work already performed and still to be done (that is, transactions processed)

The dependent regions are also considered to be resources of the system (and the most important). If one or more regions are not performing as they should, many resources probably are being misused (that is, pool space, I/O, CPU time, and the like), and throughput will suffer. The region displays are needed to analyze the actual work being done and the load balance. STAT and STATR are meant to be used only to check status, and the STAT/STATR information is very condensed to show 15 regions at a time.

The warning indicators generally suggest the use of another display. For example, if a region is in a PI wait (WARN=W-PI), the PI display shows what resource is needed and what region is holding it. However, these are warnings only, not necessarily actual problems. It is natural to have waits, so it is only when such a condition continues (through several refreshes of this display) that further analysis is indicated. If it continues, use DREGN (Program Isolation Activity area of display) or the IRLM display service to see if the condition is valid for the program being processed or a real problem. Workload Analyzer service DWAIT is an effective service to use to determine if this is a persistent problem.

The NOBK indicator is of special value when the system is in trouble and one or more regions should be cancelled. If this indicator is on, a cancel causes IMS to come down also. Previously, there was no way to know if regions were in this state.

Although the dependent regions are highly important, other resources also can be critical. If the PI pool has insufficient space, dependent regions canabend with a U775. By checking for the THRESHOLD warning on the percent ALLOCATED field regularly, steps can be taken to reduce the utilization before it causes a problem.

The line showing stopped resources on the STATR service points out unexpected conditions. If any of these values is higher than expected, the /DISPLAY STATUS command allows further analysis.

The lower part of the STAT/STATR display shows the overall performance of the system:

- How many transactions have been processed since restart?
- How many transactions are currently queued for processing?

These two counts are the true indicators of how well all resources are being used and if they are sufficient to handle the load. They can be analyzed only in comparison to what is expected of a particular system. It is possible to calculate the rate at which transactions are being processed by making two observations, calculating the difference, and dividing by the time difference. If this is done regularly at different times of the day, a very valuable system profile can be built. This profile can be used to spot deviations or slow degradation and to monitor how the system is handling a variable or increasing workload. The rate of transaction arrivals can be calculated by tracking the differences over time in the sum of the current queue count (Q'D) and total transactions processed (PROC). The Resource Monitor monitors of transaction arrivals and transactions processed do this automatically.

See Part 3, “Monitors,” in the *MAINVIEW for IMS Online – Monitors and Traces Reference Manual* for more information about:

- Monitoring paging (PAGE)
- CSA utilization (CSAUT)
- CSA fragmentation (CSAFR)
- SIO activity (SIO)
- EXCP event (SYSIO and DLIO)
- Logical channel busy (CHBSY)

To obtain a broad perspective of the system workload and to determine where a transaction is spending most of its time, use the Workload Analyzer workload wait services (MWAIT/DWAIT). These services identify possible bottlenecks and can provide sufficient information to make a correction. Otherwise, use one of the Resource Analyzer services to analyze the problem further.

## Dependent Region Monitoring

The regions display (REGNS) can be used for regular monitoring of the regions. REGNS (described in Chapter 12 on 115) contains information about the current activity in each IMS dependent region. From this display you can see:

- Which application programs are active
- How much work they have done (message queue and database access)

You can then use the DREGN display to see:

- Approximately how much work the application programs still have to do (how many transactions of this type are still queued)
- The specified processing limit

If you recognize a problem, check the logical terminal name in either the REGNS (Summary View) or DREGN display to identify the user who entered the IMS transaction being processed.

If an application program remains in a region for a suspiciously long time, a loop may be indicated. Use the REGNS Message View, DL/I View, and DB2 View to look at message queue and database activity to help locate the problem area. If the number of input messages queued (QUEUED) or dequeued (M-DEQ) is large, perhaps only the processing limit definition (PRLIM) needs to be changed.

Response time to users can be investigated by using REGNS or DREGN to view transaction elapsed time (TRN-ELAP or ELAPSED). An increase in this time is a danger signal indicating performance degradation.

Often information in a regions display indicates possible problems that can be checked out more thoroughly with other displays. For example:

- If a program loop is suspected and the total database calls (TOT shown by the DL/I view of REGNS) is high, more specific information on the types of calls being made can be seen in the DLIST or DREGN display.
- If a large number of program isolation waits is shown by the Program Isolation Activity area of the DREGN display, look at the PI service display.
- If regions are often idle, the distribution of input transactions in the class queues can be investigated in CLASQ.
- If scheduling problems are suspected, the number of scheduling failures and their causes can be investigated in the scheduling display (SCHED).

Program isolation is an important automatic feature of IMS to avoid interference between programs in database access and updating by enqueueing on database records. The Program Isolation Activity area of DREGN shows the total enqueues, dequeues, PI waits, and the number of current enqueues for each program currently processing. From this display, it is possible to see if one program is using all the resources (excessive total or current enqueues) and impacting other programs by causing PI waits.

The DL/I View of REGNS or the DL/I Call Activity area of DREGN can be used by the database administrator (DBA) or anyone concerned with application program performance. For example, these services can be used with a test system to monitor the activity performed by a new application program.

The information shown by the System Activity area of the DREGN display applies to dependent region OS/390 data. The data elements describe the region, not just the current IMS application (for example, in a message processing region where many transactions are processed by various programs). This information identifies whether the corresponding IMS task is currently executing in the dependent, DL/I, or control region (some processing, such as database access, usually occurs in the DL/I region). It shows each address space and its OS/390 dispatching priority and position in the dispatch queue.

The System Activity area of DREGN also shows the current swapping status of the region. IMS dependent regions are marked nonswappable when they are started. A small number of swaps can occur before this is completed. In addition, the total elapsed time each region has been up is shown in hours:minutes:seconds, and the total CPU time and SRB time is shown in seconds. These figures show how well the processing load is balanced between the regions.

The Paging Activity area of DREGN indicates the amount of paging DREGN in each of the dependent regions. The number of page-ins, page-outs, and reclaimed pages are displayed for that time interval, for VIO, and for the common area (CSA and LPA - no pages out). You may want to request this display at regular intervals with the CYCLE SETUP option from the Primary Option Menu to calculate the differences in the amount of this activity at various times of day.

The region transaction profile shown by the PSB/Transaction area of the DREGN display provides additional information about current transactions being processed, including the number currently queued and the total already processed since IMS restart. This information shows possible imbalances in region, class, and transaction assignments. The average length of all input messages received with a transaction code (AVG LENG) is calculated by the IMS queue manager. Tracking these average lengths for high-volume transactions can determine the optimal lengths for the short and long queue data sets (see “Analyzing Queuing” on page 6).

For more detailed analysis of transactions, workload trace (MTRAC/DTRAC) can be used to see the number of calls and database I/Os. A detail trace is useful when testing new application programs to see if they conform to specifications in the number and types of calls.

---

## Analyzing Databases

This section describes OSAM, VSAM, and fast path buffer pools. It also includes information about Hit Ratios and VSAM Hiperspace.

### OSAM Buffer Pool

A subpool concept similar to VSAM is employed in which the buffers in any subpool are all the same size. A database is assigned at open time to the subpool with the smallest buffer size in which its blocks fit, or for the specific subpool designated in the DFSVSMxx member by the database administrator.

To further improve the speed of searches to locate records in a subpool, the buffers are chained off an array with the same number of entries as buffers. The particular anchor point is determined by hashing of the DMB/DCB/RBN. This substantially reduces the number of buffers that must be examined. Because an increase in pool size does not cause a comparable increase in the number of buffers to be examined, this is particularly effective for large subpools. This allows increase of the pool size as long as the paging rate stays down.

Finally, the method of selecting a buffer to be reassigned (buffer steal) has been altered. During the selection process, each buffer in a subpool is considered to be at 1 of 11 levels (0-10), depending on the work necessary to free it. For example, empty buffers are assigned to level 0 or 1, buffers that are not currently available are at level 9 or 10. Then the subpool is divided into limited search groups with the number of buffers in each group being twice the number of scheduled regions (or one during emergency restart).

A complicated progressive search algorithm is then used to select a buffer, if one is available. The factors considered are the buffer level, the previous owner (the requestor's own buffers are preferred), the least-recently-referenced order of the buffers, and the length of the search. The net result is to reduce the search time even in large pools. The BUFFERS STOLEN and TOTAL SEARCHED by level can indicate the success of this algorithm. Levels 0, 2, and 4 show steals of the requestor's own buffers; levels 1, 3, 5 belong to another.

To make optimal use of these features, you may need to modify the database DBDs and an unload/reload. In particular, the block sizes should be as close to the defined buffer sizes as possible without exceeding it.

Once the block sizes of the IMS databases are known, the number of buffers and buffer size of the subpools can be determined. At data set open time, the first subpool whose buffers are large enough to contain the blocks is assigned. It may be possible to use this feature to separate data set blocks into different subpools. In particular, the index blocks can be isolated from data blocks. However, you must closely investigate the desirability of this.

Once you have chosen the buffer lengths of the subpools, determine the number of buffers. MVIMS can be useful in evaluating database performance. The number of OSAM WRITES-STEAL per second is a good indicator for optimizing the number of buffers in a subpool. If it is very low, there probably are too many buffers, which wastes memory and may increase the paging rate. If it is high, the number of buffers probably should be increased. STEAL WRITES are especially expensive because the buffer handler may issue an IMS log tape WRITE AHEAD call to guarantee that database changes have been physically written to the log before actually writing to the data set. This can result in two additional waits before the buffer can even be reassigned.

## VSAM Buffer Pool

The IMS VSAM buffer handler/pool consists of 0 to 255 subpools of fixed length buffers. At database open time, the smallest subpool that can contain the control intervals is assigned, unless the database administrator has assigned a specific subpool for that database.

Although this fixed length buffer subpool technique (similar to current IMS OSAM buffering) eliminates pool compactions, the buffers in each subpool are chained in a least-recently-referenced use chain. This causes all buffers to be examined during searching. When a buffer is needed, VSAM selects the least-recently-referenced. If the buffer contains data that has been modified, it is first necessary to write it out. To improve performance, VSAM writes are deferred until space is needed or an explicit request is made to purge a user's buffers.

The number of WRITES (VSAM initiated) per second should be monitored to see if sufficient buffers have been allocated. Such writes are expensive because they may require a log tape WRITE-AHEAD call. This can result in two additional waits before the buffer can be used.

To prevent this condition, IMS employs a BACKGROUND WRITE feature, which can be turned on or off at system initialization (OPTIONS statement). This is a low priority IMS ITASK that runs when VSAM notices that the next buffer to be freed on the use chain has been modified. It forces VSAM to write out a specified percentage of buffers from the least-recently to the most-recently-referenced. The NUMBER OF TIMES BACKGROUND WRITE INVOKED per minute may also give an indication if there are enough buffers.

**Note:** Various internal traces may introduce overhead to DL/I calls and buffer handling. The status of these traces may be observed in the Resource Analyzer service and may be turned on or off by the IMS /TRACE command.

The fixed length buffers make it possible to separate control intervals by subpool. Of special importance is the separation of index blocks from data blocks.

IMS supports an index-only VSAM buffer pool. This allows separation into separate pools data and index control intervals of the same size.

IMS supports buffer pools that are dedicated to specific databases. This can be used to give preferential treatment to these databases or to isolate high activity databases.

## Hit Ratios

Buffer pool hit ratios can be used to evaluate how well the buffer pools are performing. The hit ratio is the percentage of buffers that were found in the pool without needing to access external storage. Generally it is recommended that data buffer pools have a hit ratio of 60 or higher and index buffer pools have a hit ratio of 80 or higher. These hit ratios are available with the following services:

Table 1. Services Supporting Hit Ratios

	<b>Analyzers</b>	<b>Monitors</b>
OSAM	DBST	DBHIT
VSAM	VSST	VHIT

## VSAM Hiperspace

When defining your VSAM buffers to IMS, you can request that VSAM allocate Hiperspace buffers in expanded storage to augment your virtual storage buffers. Buffers that normally would have been washed out of the pool due to the least-recently-used algorithm will be migrated instead to Hiperspace buffers. This allows you to have more buffers without increasing your virtual storage requirements.

The number of successful reads from Hiperspace is a measure of the number of reads with I/Os that were saved by using Hiperspace. The number of unsuccessful reads represents the number of Hiperspace buffers that were stolen because of a shortage of expanded storage. To get the full benefit of your Hiperspace buffers, the number of successful writes to Hiperspace should be less than the number of successful reads from Hiperspace. If the number of successful writes is greater than the number of successful reads, buffers are being written but never read.

The Hiperspace hit ratio shows the portion of the VSAM hit ratio contributed by Hiperspace. In other words, your VSAM hit ratio would be smaller by the Hiperspace hit ratio amount if you had not defined Hiperspace buffers. This is displayed on VSST and monitored by HPHIT.

## Fast Path Buffer Pool

Buffers from the fast path buffer pool are fixed per dependent region that requires buffers. This is determined by the NBA= keyword in the dependent region startup JCL. If the buffers are not available in the CICS environment, the region fails to connect to IMS (either DBCTL or IMS DB/TM). Altered buffers are not written to the area data sets until synchronization point.

The keywords that define the buffer pool in the control region JCL are:

- BSIZ, which specifies the size of the largest DEDB Control Interval (CI)
- DBBF, which defines the total number of buffers
- DBFX, which defines how many of these buffers are to be used as reserve buffers

These reserve buffers act as a cushion at synchronization point time to allow the asynchronous output threads some time to complete writing the buffer while the next transaction is processing. If the output threads are being delayed or there are not enough of them (OTHR keyword), a wait occurs. IMS does not keep global statistics on this wait condition and it must be analyzed by examining the IMS PR report TRNFP or the batch fast path log analysis utility (DBFULTA0).

**Note:** A buffer is fixed in the buffer pool for each active area that has a sequential dependent part defined (SDEP).

Dependent region allocations of buffers are determined by the NBA and OBA keywords. The normal buffer usage (NBA) should be large enough so that overflow buffers (OBA) are seldom used. IMS reserves one set of overflow buffers out of the buffer pool that is equal to the largest OBA specification in all of the dependent regions. This means that only one dependent region can use these buffers at a time causing serialization; for example, contention for overflow buffers. Detailed analysis of the dependent region buffer usage is performed by the IMS PR report TRNFP or the IMS batch utility (DBFULTA0).

The FPBST display assists in online determination of buffer usage. By refreshing this display and monitoring the buffers in use by region, you can detect an NBA specification that is too small; for example, in the region detail display the USED is consistently more than the NBA.

You can tune the OTHR and DBFX values by monitoring the number of IDLE OUTPUT THREADS and FIXED BUFFERS AVAILABLE. If the idle threads are consistently 0, they should be increased. If the fixed buffers available are consistently a small percentage of fixed buffers, there is a strong possibility that waits will occur for buffers. To overcome waits due to insufficient output threads or available fixed buffers, increase the number of pre-fixed buffers (DBFX) and/or the number of output threads (OTHR).

## Points Common to All Buffer Pools

In evaluating IMS system performance, you should determine the amount of physical I/O per second. Although it is possible to break the activity down by subpool with Resource Analyzer and Resource Monitor services, the PERFORMANCE REPORTER offline Database I/O report can be used to determine activity by DMB. Using the DC Monitor, it is possible to determine activity by data set; however, even this does not show contention between data sets on the same device or activity to a data set spanning multiple volumes (devices). This information can be obtained by using CMF MONITOR or MAINVIEW for OS/390, both BMC Software products, or the IBM Generalized Trace Facility (GTF). These give a better indication of the actual use or contention of devices and paths. This also can be estimated by analyzing the database change records. However, this method is incomplete because retrieve type calls are not logged and a special analysis program would have to be written.

The number of events per second reported, using the techniques in the previous paragraph, do not take into account the number of transactions that generated the activity. The number of events per second could be re-expressed in terms of events per PURGE by using either the VSST global display or DBST display service. A more detailed investigation can be made with the MVIMS transaction processing report.

These calculations should be made at different times (when the system is busy and when it is not busy). Investigate changes observed over time that show a trend. In particular, an increase in READS/SEARCHES per PURGE shown by the VSST global display or DBST display service may indicate the searching of an overflow chain. This could be an HISAM/HIDAM-INDEX overflow chain or an HDAM synonym chain. If so, the database involved should be determined and reorganized.

## Suggestions:

- HDAM is probably the most efficient IMS access method; however, long synonym chains must be watched. If encountered, investigate:
  - Randomizer - the most widely used and efficient randomizer is the WORLD TRADE (DFSHDC40).
  - Insufficient RAPs for the number of records (resulting in long synonym chains).
  - Too many RAPs per block for the record size.
  - Not using the BYTES option. This is especially devastating after a reorganization.
  - The advantage of distributed free space.

HISAM is the best for sequencing small records.

- In VSAM, watch CI/CA splits. It may be possible to use the VSAM KSDS options to reduce these splits.

HIDAM seems to be the most used or abused.

- The HIDAM INDEX is nothing more than a HISAM database and suffers the same problem of long overflow chains or CI/CA splits.
- If much sequential processing is done, it should have both forward/backward pointers at the root to reduce index access.
- If many dependent segments are added to a record, it may help to specify distributed free space.
- IMS data sets should be allocated by cylinder as a single extent. If necessary, they should be reorganized periodically to a single extent. This can be done with IEBGENER for OSAM. EXPORT/IMPORT or REPRO can be used for VSAM (this reorganizes a KSDS also). However, this is not a substitute for a complete IMS database reorganization.
- Watch the placement of data sets to prevent an over- or under-utilization of a device, control unit, or channel. Try to avoid placing two very active data sets on the same device. Also, watch for an excessive number of alternate tracks, especially when these tracks are associated with a high usage data set. This situation tends to reduce seek time.
- Write check should not be specified for IMS databases. There is no integrity exposure because the data sets can be recovered using the IMS utilities and log tapes.
- For VSAM, SPEED should be requested because IMS does not use VSAM recovery. This speeds up initial loads.

---

## Analyzing IMS Internals

This section describes latches, logs, pools, and program isolation analysis.

### Latches

IMS latches, like OS/390 locks, are used to protect the integrity of certain resources in a multi-programming environment. A certain number of conflicts are to be expected and show that the latches are performing a needed function. However, excessive conflicts may indicate a malfunction in system performance that you should investigate.

In general, latches are held for only a very short time, which reduces the number of conflicts. If something occurs that increases the time a latch is held, the probability that another task will need the protected service before the task completes also increases. The main causes of such delays are page faulting or faulty dispatch priorities. If a latched routine has to page fault through a pool, the execution of any other task waiting for that service or resource can be blocked, causing a degradation in IMS performance.

### Suggestions for Relieving Latch Conflicts:

- Investigate the paging rate of the system and consider page-fixing the affected pools (use the LATCH service to see where conflicts are occurring most frequently).
- If latch conflicts are appearing for DMB user routines, check these routines for waits or possible page faults.
- If the conflicts are frequent for certain OSAM buffer subpools, review the definition of the number of subpools and the number of their buffers to reach a better balance between the allocation of the available space and the actual usage of the buffers.

### Logs

Three IMS data sets are used, two online and one offline. Resource Analyzer service LOGST displays utilization information and statistics about the two online data sets and provides certain restart data.

### Online Log Data Set (OLDS)

The IMS online log data set (OLDS) can be either single or dual and contains complete log records.

IMS only writes the buffer (padded if necessary) when the buffer is completely full. If OLDS and WADS are allocated by JCL, the number of buffers for the OLDS is specified on the OLDS DD statement. If OLDS and WADS are allocated dynamically, the number of buffers is specified by the BUFNO parameter in PROCLIB member DFSVSMxx.

When an OLDS becomes full, it must be archived. Once the archive batch job is complete, the OLDS can be reused. If dual OLDS are in use, a switch is made when either OLDS becomes full.

Buffer size for the OLDS is taken from the preallocated data set. At least four or five buffers should be assigned because one of the buffers is used to read the OLDS if a dynamic backout is required.

## Write-ahead Data Set (WADS)

The IMS write-ahead data set (WADS) is preformatted and is used to write incomplete OLDS buffers. WADS does not have its own buffers; it uses the OLDS buffers. The OLDS buffers are segmented in 2K segments. Any check write requests cause unwritten segments for the current buffer to be written to the WADS. This allows the log write-ahead requirement to be satisfied.

Database log write-ahead is no longer an option; it is compulsory.

Data communications log write-ahead (DCLWA) is a default option. Review it carefully. Response time can be adversely impacted if the default DCLWA=YES is allowed or specified. Response time is impacted while the transaction or message is written to the WADS, which delays processing; however, this does give complete DC integrity and should be evaluated.

IMS restart processing accesses the WADS to close the OLDS in the event of a failure.

## System Log Data Set (SLDS)

The system log data set (SLDS) is an offline data set and can be either tape or DASD. The SLDS is used for archiving the OLDS.

## Log Performance Suggestions

- Prevent DASD contention on all online log data sets; that is, separate paths, strings, devices (no shared DASD), and so on. Consider contention that can occur because of an archive utility that is executing as a batch job.
- Allocate the WADS on a low-usage device because this is most critical.
- Provide for multiple WADS backups.
- Allocate OLDS and WADS with contiguous space.
- Specify approximately 10 buffers in the archive job for both OLDS and SLDS.
- Ensure that the archive job runs at a high priority.

**Note:** The Resource Analyzer service LOGST produces a warning message if either the last OLDS is in use or the system is waiting for an archive to be performed.

The LOGST service also displays the OLDEST LCRE. This field identifies the oldest recovery point that IMS will require in the event of a restart. A very old time in this field could indicate a BMP or JBP that is not taking sufficient checkpoints and the record required for restart might already be archived.

## Program Isolation Analysis

Program isolation is the IMS function that allows many application programs to access the same databases concurrently without interference. It achieves this by enqueueing on each database record as it is accessed at READ, UPDATE, or EXCLUSIVE level. Normally, each enqueue requires two QCBs (queue control blocks) from the PI pool. To improve performance, an entire record is locked by enqueueing the root segment at UPDATE level when a segment is accessed. When moving from record to record, the new root segment is enqueued at UPDATE level and the old root segment is dequeued. Dependent segments are enqueued only when modified. These enqueues remain until the modifying program reaches a synchronization point. Any other program attempting to access a segment or record that has been so enqueued must wait.

The PI service allows investigation of program isolation problems. Watch the amount of FREE SPACE in the DYNAMIC POOL closely. If more space is required and the MAXIMUM POOL SIZE has not been reached, a GETMAIN is done in subpool 241 (CSA) for the amount specified in INCREMENT. When the pool can no longer be expanded, the requesting program is pseudo-abended with a u775. All changes are backed out and the transaction is put back on the queue for reprocessing. This is an enormous drain on IMS and causes a sharp decline in performance.

However, it is not a good idea to allow the PI pool to grow uncontrollably. Once space is obtained, it is never released. Because of the space management algorithm, the total available space is constantly referenced even after most of it is free. This can increase the page faulting rate for all IMS regions. Because this is also one of the few pools that cannot be page-fixed, page faulting can be reduced only by keeping this pool small. Use the PIMAX monitor to make sure that one region is not using this pool too much. If it is caught early enough, a region can be stopped before it fills up the pool.

The bottom portion of the PI display gives the number of enqueues each dependent region currently holds. This is broken down by level of enqueue:

- READ
- UPDATE
- EXCLUSIVE

More importantly, it is possible to see if a program is in a PI wait, what resource it is waiting for, and which program is holding the resource. If there are frequent PI waits, serious attention should be given to the application system design or database design. This information is especially valuable in determining the checkpoint call frequency needed in BMPs and JBP. You can possibly reduce contention by changing the processing option to GO or EXCL to bypass PI. Some cases may require multiple database PCBs with different PROCOPTs. You can use explicit program enqueue/dequeue control using Q command codes to control simultaneous access. In a few cases, the problem may not be solved without a complete system redesign.

In addition to lengthening program execution time, PI waits may lead to deadlocks. Although not fatal, IMS must pseudo-abend one of the programs. All of this transaction's changes are dynamically backed out and the transaction is put back on the queue for reprocessing.

**Note:** IRLM can be used to control database contention in place of program isolation.

## Pools

There are two kinds of pools: CBT and non-CBT. The POOLC service displays CBT pools. The POOLS service displays the status of the non-CBT pools. The DPOOL service displays detailed information about non-CBT pools.

### CBT Pools

Although you do not have direct control over most CBT pools, you can control DPST and SAP. You should consider the following:

#### DPST Pool

The DPST pool holds the dependent region partition specification tables (PSTs). DPST is defined by the Stage 1 system generation, (`IMSCTRL MAXREGN=` or `DFSPRRGO PST=`); this definition can be overridden by the JCL parameter `PST=`. DPST defines the minimum number of PSTs that IMS should hold available. Always define the average number of regions that you intend to run. Understating the number does not cause a PST shortage because the pool automatically expands and contract. However, the GETMAIN and FREEMAIN processes are an unnecessary overhead and can be avoided by making the correct size definitions.

Additionally, `PST=` defines the number of VSAM strings, which do not change. If there is a shortage of VSAM strings due to an understatement of `PST=`, the dependent region waits.

#### SAP Pool

The SAP pool is used to hold all the dynamic and other save area prefixes (SAPs). SAP is defined in the Stage 1 system generation macro (`IMSCTRL MAXI 0=` or `DFSPRRGO SAV=`; this definition can be overridden by the JCL parameter `SAV=`. This pool does not expand and its shortage causes IMS to enter selective dispatching which severely degrades performance.

### Non-CBT Pools

#### WKAP Pool

Main pools contain the DFSISMN0 control blocks and a general work pool, WKAP. MVIMS displays only the WKAP portion. IMS creates temporary work pools from WKAP. The size of a temporary work pool is defined by `DFSPRRGO WKAP=`; this definition can be overridden by IMS control region JCL parameter WKAP.

#### QBUF Pool

The QBUF pool is the central storage area used as I/O buffers for the three types of message queue data sets: SHMSG, LGMSG, and QBLKS. The IMS storage manager module DFSISMN0 allocates the storage for the QBUF pool during IMS initialization. The IMS message queue buffer manager module DFSQBFM0 manages the QBUF pool during transaction processing. A well-tuned QBUF is essential to fast transaction response time because all inbound and outbound message traffic must pass through QBUF.

### **QBUF Allocation:**

The size of the QBUF pool is defined during Stage 1 system generation by the MSGQUEUE macro. You can use the QBUF parameter in the DFSPBxxx parmlib member or in the IMS JCL EXEC statement to override the number of QBUF buffers at execution time. You can use the QBUFSZ parameter in the EXEC statement to override the size of the buffers. To page-fix the QBUF pool, set EXVR=Y in the EXEC statement for the control region JCL.

### **QBUF Internals and Associated MVIMS Displays:**

IMS caches as many messages in the QBUF pool buffers as possible before it attempts a buffer steal, a process that requires I/O to one of the message queue data sets SHMSG, LGMSG, or QBLKS. A buffer steal is not attempted until all the buffers have been used. DFSQBFM0 attempts buffer steals from the stealable buffers first. A buffer is stealable if it is not currently involved in I/O and it has no IWAITs. To steal a buffer for use by a different message, DFSQBFM0 must first write the existing buffer contents to one of the message queue data sets. If a buffer is currently involved in an I/O or otherwise owned by another process (busy, in other words), an IWAIT is always required before the buffer can be written and stolen. For that reason, access time to a stealable buffer is much faster than access time to a busy buffer.

When a buffer is being stolen, the QBSL latch is held. MVIMS displays IWAIT data for the QBSL latch in the LATCH Summary display. The latch may be held without causing an IWAIT. However, if there are no stealable buffers when an attempt to acquire the QBSL is made, an IWAIT for QPWTBFR (wait for an available buffer) is incurred before the latch can be acquired. The number of IWAITs for QPWTBFR is provided in the QUEST Statistics analyzer display in the WAITS FOR AN AVAILABLE BUFFER field. If the value in the field is not zero, all QBUF buffers are busy and none are stealable.

Area 1 of the DPOOL analyzer display for the QBUF pool shows the number of stealable buffers (STEAL) and busy buffers (CURR). The POOLS analyzer display shows the number of busy buffers (CURR), and the value is the same as the CURR value in the DPOOL display. The POOLA monitor monitors the QBUF pool for high contention on its buffers. Contention is defined as buffers that are busy (the CURR value in DPOOL and POOLS) and not stealable (the STEAL value in DPOOL). Using this information, you can see what portion of the QBUF pool is busy (CURR) and what portion is used but not busy (STEAL).

### **QBUF Tuning:**

If your system is not real-storage constrained, your QBUF pool should be set large enough to ensure minimal I/O to the message queues. In general, the RECLENG parameter in the MSGQUEUE macro should be set so that I/O activity to SHMSG and LGMSG is evenly split. The split is determined by message segment lengths and the distribution of message arrival between message segments shorter than LGMSG LRECL and longer than SHMSG LRECL. I/O to the message queues may occur because the mix of arriving message segments uses up one or the other section of the QBUF pool buffer even though the size of the QBUF pool seems large enough. You should carefully balance the path length of buffer searching and I/O to the message queue data sets. If you overallocate the number of buffers, the buffer search path length and CPU cycles will increase. If you underallocate the buffers, I/O will increase. One tuning technique to reduce storage requirements and CPU cycles for QBUF buffer operations is to decrease the number of buffers until the QUEST Statistics analyzer display shows increasing IWAIT instances for message queue I/O.

## PSBW, PSB, and DMB Pools

These pools are managed by DFSISMN0. If `LS0=S` is specified, the PSB pool is split into two parts. The POOLS service shows the parts as DLMP/PSBC in CSA and DPSB/PSBD in private for the DLISAS address space. The DMB pool is shown as DLDP/DMBP in CSA. You can use the DPOOL service to get details for each pool. Each pool has associated monitors that can be started (such as PSBP and DMBP).

During transaction scheduling, IMS allocates space from these three pools to accommodate the required PSB and DMB. If space allocation fails in any of these pools, then IMS tries to make space available by purging not-in-use control blocks. If space is still not available, then scheduling fails and IMS tries to schedule the next eligible transaction. The dependent region waits if none can be scheduled. For more information, see “Pool Utilization” on page 10.

DMBW is the DMB work pool. Although it is not used during transaction scheduling, it is used later by the DL/I delete/replace action module (DFSDCDC0).

The resident PSB and DMB are loaded at IMS startup time into storage outside these pools. The resident PSB is copied into the PSB pool when needed. The resident DMB does not need to be copied.

The PSB pool holds the PSBs on a most-recently-referenced basis. It is defined in the Stage 1 system generation (`BUFPOOLS PSB=`, `SASAPSB=` or `DFSPRRGO PSB=`, `CSAPSB=`, `DLI PSB=`); this can be overridden by the JCL parameters `PSB=`, `CSAPSB=` and `DLI PSB=`. The IMS default for splitting the PSB pool (80/20) is reasonable, so let it default.

The PSB work pool (PSBW) holds various DL/I work areas for the PCBs such as index, SSA, and SPA. It is defined in the Stage 1 system generation (`BUFPOOLS PSBW=` or `DFSPRRGO PSBW=`); this definition can be overridden by the JCL parameter `PSBW=`.

The DMB pool holds the DMBs on a most-recently-referenced basis. Space shortage causes some DMBs to be closed and flushed. The DMB pool is defined in the Stage 1 system generation (`BUFPOOLS DMB=` or `DFSPRRGO DMB=`); this definition can be overridden by the JCL parameter `DMB=`. The DMB work pool is defined in `DFSPRRGO DBWP=`; this definition can be overridden by the JCL parameter `DBWP=`.

## CESS Pool

The CESS pool allocates external subsystem communication control blocks, such as the DB2 attach facility.

## MFBP Pool

The MFS buffer pool, MFBP, accommodates MFS control blocks. However, MFSTEST control blocks are not taken from this pool; they are taken from the CIOP pool. MFBP size is defined in the Stage 1 system generation, (`BUFPOOLS FORMAT=` and `FRE=` or in `DFSPRRGO FBP=` and `FRE=`); this definition can be overridden by the JCL parameters `FBP=` and `FRE=`. This pool is allocated from extended CSA.

## CIOP, HIOP and RECA Pools

The communications I/O pool, CIOP, is used as message buffers between VTAM/BTAM and the IMS queue manager. RECANY has its own CBT pool, RECA. CIOP contains only the output buffers and EPCB blocks, described below.

A high communications I/O pool, HIOP, is allocated from IMS control region extended private area to expedite the CIOP usage. This pool is defined in the Stage 1 system generation (macro `BUFP0OLS COMM=` and macro `COMM RECANY=` or in `DFSPPRG0 TPDP=`); this can be overridden by the JCL parameter `TPDP=`.

Because the MFSTEST work area is taken from the CIOP pool, consider the sizes of both MFSTEST and the CIOP pool. MFSTEST size is specified in IMS JCL `MFS=`.

## EPCB Pool

The EPCB pool holds the Fast Path PCBs. EPCB storage shortage causes the Fast Path transaction scheduling to fail without any indication to you.

---

## Analyzing the System

This section describes dispatching and real storage.

### Dispatching

The dispatcher statistics display (DSPST) gives an overview of the status and activity of both OS/390 and IMS dispatching.

In the first section, the IMS control region, the DL/I SAS, and the dependent regions are listed with pertinent OS/390 data. The default dispatch priority is set in the supplied PROCs at 239. The dependent regions should run at a slightly lower priority. To assist IMS in its attempt to balance the activity, the priorities of the dependent regions normally should be equal or quite close. This should be changed only if specific processing characteristics of the installation require special consideration.

The SRM parameters of domain and performance group/period and the current swap status can give an indication of how OS/390 functions are affecting the performance of the IMS regions. As a general rule, only one performance period should be defined for the IMS performance group(s), because the IMS regions should be allowed to do their job without excessive interference from the OS/390 SRM facilities.

The IMS dispatcher is responsible for the IMS internal multi-tasking by creating and dispatching ITASKS. Each ITASK is associated with an ECB (event control block) that is posted when work is to be performed for a particular function and with a SAP (save area prefix) that controls the IMS resources associated with the task (for example, a save area set for the registers of all the invoked IMS routines).

There are pre-assigned SAPs for activity associated with each dependent region, logging, and the like. Dynamic SAPs are used for all terminal I/O activity in IMS/DC. The number available depends on the specification in the MAXIO statement during the generation or the SAV parameter at execution. Because each SAP is associated with a GETMAINED area in CSA of over 1200 bytes for the save area set, this number should be optimized to conserve space. You can accomplish the optimization by reducing the number of dynamic SAPs each time IMS is brought up (but only until a few occurrences of TOTAL ITASKS WAITED FOR A SAP appear). After this point, performance might be affected by further reduction.

ITASKS are created and dispatched for all terminal I/O activity, and destroyed upon completion. By subtracting the number of ITASKS created from those dispatched, an approximation of the number of IWAITS can be found. This is an indication of the total IMS I/O activity. For example, a dependent region must IWAIT for database access. This number also can be calculated per second and used as a standard performance indicator to be checked over time.

## Real Storage

Probably the greatest single cause of poor IMS performance is an overcommitment of real storage. This results in a high demand paging/swapping rate. Not only does it waste I/O, but also CPU. These effects are particularly devastating to a lightly loaded IMS system.

On such a system, the IMS modules constantly page in and out. Because of the design of IMS and OS/390, a region is effectively dead during page fault processing. Once it regains control, it usually immediately page faults again because of the very large working set size of IMS.

The IMS latches and commonly referenced database records are of particular concern. The only thing that can be done to reduce such interference is to reduce the page faulting in IMS. While it helps to reduce the non-IMS jobs that run concurrently, there may not be any alternative to page-fixing a portion of IMS.

The RS service allows the investigation of real storage usage. Because IMS is such a heavy user of CSA and LPA, these are also included. It gives a complete breakdown of the status of the usable page frames in the system:

- Total
- Pageable
- Fixed
- Long-term fixed

In an identical form, the status of the page frames used by IMS are given.

---

## Chapter 2. User Techniques for Monitors

This section provides assistance in using the functions available with the data collection monitors. This section is not a step-by-step description of all the uses of these functions, because installations have varied requirements depending on their processing load and available resources (which may vary over time) and because there are so many services available.

The built-in flexibility of the monitor services allows you to determine which items are important to observe, when to observe them, and how long to observe them. This flexibility also allows you to vary these specifications easily when needed.

The monitor services offer continuous monitoring, early warnings, graphics, and rate calculations. They can be used as:

- An operations monitor of current IMS events
- A performance analysis tool to assist in tuning the system to use the available resources better and to plan for the future

With the monitor services and the timer facility that drives them, these tasks can be done automatically and selectively.

The following sections describe uses of the monitor services by different people within the IMS organization. This is not a complete list and many of the services could be of value to other people within the organization. The technique descriptions use sample block requests delivered with the product. The descriptions are grouped by the following users to present the various ways that an organization can use MVIMS:

- Master terminal operator
- IMS manager
- IMS performance analyst and system programmer
- Database administrator

---

## Master Terminal Operator (MTO)

The master terminal operator is responsible for controlling the system and keeping it up and running smoothly. Responsibilities include controlling the resources and solving operating problems. The sooner problems are detected and the causes analyzed, resolving or minimizing the effect, the better service IMS can give to the user. The automatic warning feature of MVIMS can relieve the MTO of the chore of continuously monitoring all the different aspects of the system. A set of standard monitor requests can be defined for the installation that can automatically monitor the system and inform the MTO when a potential problem (as defined for that system) is detected.

Warnings are sent automatically to the BBI-SS PAS Journal log, where they can be browsed. Warnings can also be sent to the MTO console. If MAINVIEW AutoOPERATOR for IMS is installed, warning messages can automatically invoke AO EXECs to take action or put the warning message in the AO STATUS/EXCEPTION panel.

Two types of measurements can be set up; one to monitor specific problem areas (such as the usage of a limited resource) and one to monitor general system performance indicators (such as input queue length or scheduling failures).

### BLKMTO Example

```
REQ=INQTR  WMAX=20  WLI M=5  LOG=ATWARN  WMSG=MTO
```

This sample request monitors the total input queue length every minute, automatically issues a warning message to the MTO when the queue length exceeds the specified critical threshold (WMAX=20), suspends warning messages after five are issued (WLI M=5), and logs a plot of the accumulated history data whenever a warning message is issued (LOG=ATWARN).

A long input queue is one of the first indicators that system problems exist. Something may be interfering with normal processing, causing a backlog of transactions and a degradation in response time. The critical queue length varies for each installation and the threshold that defines the warning condition (WMAX) should be adjusted accordingly. If an installation stops transactions in the queue often (perhaps priority 0 transactions waiting for BMPs), the MTO might choose to monitor only the transactions that can be scheduled (REQ=IQSCL). In a Fast Path environment where the expedited message handler is used, the monitor INQBG would be used. Other factors may influence the definition of a warning condition; for example, the submission of batched transactions, which causes a quick jump in input queue length. Warning messages for this expected condition can be avoided by setting the WIF option; for example, WIF=3 specifies that a queue length greater than the threshold must exist for three intervals before the warning message is issued. When the MTO is informed of a long input queue, other MVIMS displays, CLASQ, the history PLOT of INQTR, REGNS, DREGN, or SCHED can be used to detect the cause of the problem.

```
REQ=CSAUT  WMAX=80  WMSG=WT0  I=00: 05: 00
```

This request monitors CSA for a usage percentage. CSA is a critical resource for IMS systems with high program isolation activity. This percentage is checked every 5 minutes (I=00: 05: 00), and a message is sent to the system console if it exceeds 80 percent (WMAX=80).

```
REQ=LGMSG  WMAX=70  WMSG=MTO
```

This sample request monitors the usage of the long message queue data set and informs the MTO if it exceeds 70 percent (WMAX=70). This early warning should give the MTO time to determine the cause and take action before a full queue data set causes an IMS ABEND. For example, if an application program is in a loop and is writing invalid messages to the queue, the MTO could cancel the program.

```
REQ=@RSPC, 1  WMAX=3. 5  WIN=2  WMSG=MTO
```

This Workload Monitor sample request monitors MPP transaction response time for transactions in processing class 1. For the purposes of this example, it is assumed that class 1 comprises short-running transactions which must have a 3.5-second or shorter response time to meet service level objectives. When the average response time in an interval for class 1 transactions exceeds 3.5 seconds, a warning message is issued. To give the MTO time to react, subsequent messages are issued every two intervals (WIN=2) while the condition persists.

Transaction response time may increase for many reasons. For example, a sudden increase in the arrival rate of class 1 transactions can exceed the capacity of available message processing regions to process them, or an increase in the overall system real storage demand may cause unacceptable amounts of paging. The former case can be diagnosed by using the STAT/STATR display or the history PLOT of ARVCL. The latter case can be diagnosed by using the DREGN display (Paging Activity area) or the history PLOT of PAGE.

There are many other resources and indicators which can be measured, from system data set usage, internal pool usage, and paging, to the level of PI enqueues. Depending on installation configuration and activity, some or all of these services can be defined to automatically monitor the most critical areas. Because the number of requests and the sampling interval for each can be set and modified as needed, the amount of MVIMS activity (and resulting system overhead) can be controlled and channeled to fulfill real needs.

---

## IMS Manager

The IMS manager can use monitor services for high-level monitoring of the total system and for selectively tracking special areas.

### BLKMGR Example

REQ=PRCTR      I=00: 06: 00      LOG=ATPD      WMSG=MT0

This sample request monitors at 6-minute intervals the number of transactions processed and automatically sends an entry to the IMS log at the end of each period (10 intervals = one hour). This up-to-date, graphically presented data, with the processing rate-per-second already calculated for short- and long-term time periods, is available for online access at any time while the request is active. The sampling interval can be set to show a different spectrum, from seconds to hours. If this request is started before transaction processing begins, the total count field shows the total number of transactions processed that day.

REQ=ARVPR, ACC+      LOG=ATPD      RANGES=0, 10, 60, 300      WMSG=MT0

This sample request monitors the pattern of transaction arrivals for a group of programs (ACC is a qualified name), automatically sends a plot display record to the IMS log at the end of each period (10 intervals), and displays the distribution of the number of arrivals per interval in the ranges 0 to 0, 1 to 10, 11 to 60, 61 to 300. In this case, the arrivals are monitored for a program group; however, arrivals may be monitored for a transaction, transaction group (qualified name), processing class, program group (qualified name), or application program name. The form of the request depends on installation naming conventions. The IMS manager can use this information to discuss system usage and performance and have statistics for the period in question, not just totals or averages for a full day.

The frequency distribution produced indicates how often a certain arrival rate was measured: no arrivals in one minute, 10 per minute, 60 per minute, 300 per minute or over. The highest count ever measured is shown.

REQ=@RSPT, AR+      LOG=ATPD      RANGES=2. 5, 5, 7. 5, 10      WMSG=MT0

This Workload Monitor sample request monitors the transaction response time for a group of transactions (AR is a qualified name), automatically sends a plot display record to the IMS log at the end of each period (10 intervals), and displays the distribution of transaction response time in the ranges of 0 to 2.5 seconds, 2.51 to 5 seconds, 5.01 to 7.50 seconds, and 7.51 to 10 seconds. In this case, transaction response times are being monitored for a transaction group; however, response times also may be monitored by processing class, LTERM, region, program, or user. Qualified names are allowed for every option except class. The form of the request depends on installation naming conventions. The IMS manager can use this information to discuss system response time and service level objectives with users and can generate statistics for any period in question, not just totals or averages for an entire day.

---

## IMS Performance Analyst and System Programmer

The person in charge of IMS tuning has a very complex task: determining the best use of available resources to maximize user service. The large number of variables to be monitored, the many parameters that can be adjusted, the interaction of the various internal IMS functions such as queueing and scheduling, the effect of operating system constraints, the continual variations in processing load and applications mix, and the number and size of the resources to be controlled all increase the difficulty of understanding and tuning the IMS system. Offline reports of summarized data often do not give the precise information needed to analyze the causes of current problems or to detect potential bottlenecks in resource usage.

With the monitor services, the performance analyst can selectively monitor only the information currently needed. The amount of output produced and the cost of producing it can be controlled.

Many different resources and performance indicators can be measured as needed, such as pool usage, I/O rates, program isolation activity, input and output queues, paging, and CSA. Many of these can be measured either globally or selectively which allows the collection of exact information for specific problem areas.

### BLKPERF Example

```
REQ=DBSTL, 2      I=00: 10: 00      WMAX=100      LOG=ATWARN, ATSTOP  
START=10: 30: 00  STOP=11: 40: 00
```

This sample request monitors the number of buffer steal writes performed between 10:30 and 11:40 a.m. in the OSAM database buffer subpool 2. The graphic history can be inspected online at any time after 10:30 until the request is specifically purged. In addition, the requestor is informed at the terminal whenever the number of steal writes exceeds 100 in any 10-minute interval. The history plot is written automatically to the BBI-SS PAS Image log for later offline analysis whenever the warning threshold is exceeded and also at 11:40 when the measurements for this request are discontinued.

Several different I/O counters can be monitored, either by IMS function, by using the MFSIO service or the DBSTL service, or on a system level, such as start I/Os by unit or paging rates.

```
REQ=PSBP      I=00: 00: 30      LOG=ATPD
```

This sample request creates a detailed graphic history of PSB pool usage. The percent allocated is measured every 30 seconds and a plot record written to the log every 5 minutes (30 seconds x 10 intervals = one period).

This is just one example of how the performance analyst can use MVIMS to selectively monitor special problem areas whenever needed. Other, more generalized requests can be set up to always be active, but only write BBI-SS PAS Image log records when warning conditions occur. Analysis of these records shows areas that need more detailed study.

The WARNINGS WRITTEN field on the Timer Facility Statistics display (Primary Menu Option 5) shows whether any warning conditions were detected.

```
REQ=SI0, 158    RANGES=10, 60, 300, 1500
```

This sample request tracks the number of successful start I/Os issued to unit 158, which is used by IMS (for example, a database). In addition to the graphic display of each minute's activity and the calculation of rates-per-second, a frequency distribution is accumulated. The range limits defined are for the activity count within an interval, in this case one minute. This corresponds to a distribution of rates-per-second of 1 every 6 seconds, 1 per second, 5 per second, and 25 per second. The last range limit shows the highest value measured. If IMS is run with LS0=S, DLIO can be used for data sets allocated in DLISAS.

REQ=CLASQ            I=00: 05: 00    LOG=ATINTVL

START=09: 00: 00    STOPCNT=50

This sample request writes an IMS log record of the Class Queuing display every 5 minutes until 50 are created. This automatic logging of an informational display can create an audit trail of detailed data at a regular interval within a certain time period. These records can be printed with IMRPRINT and analyzed at any time.

---

## Database Administrator

The database administrator often is responsible for monitoring and controlling the performance of the application programs. With several of the monitor services, an automatic process can be activated, which checks for adherence to installation standards and good programming practices. The effects of new applications on the total system and the performance and usage patterns of the application itself can be monitored.

### BLKDBA2 Example

```
REQ=DBTOT(MSGRGN01)    WMAX=20    I=00: 00: 01
```

This sample request for service DBTOT checks the level of DL/I activity occurring in a region. For example, if the first region (always started as job MSGRGN01) is only supposed to process fast transactions issuing a small number of DL/I calls, this request checks every second and issues a warning message to the BBI-SS PAS Journal log if the program being executed issues more than 20 database calls per scheduling.

```
REQ=PI ENQ BMPRGNXX    WMAX=250    I=00: 00: 10
```

This sample request checks every 10 seconds and automatically writes a warning message to the BBI-SS PAS Journal log if the BMP in this region (when it is active) ever has more than 250 outstanding PI enqueues. The DBA can look at the PI display whenever a warning is received to see if this level of PI activity is causing conflicts with other regions at that time. The plot of this request gives a full history of the enqueue levels reached between CKPT calls. This information allows fine tuning of the frequency of CKPT calls needed in that BMP.

```
REQ=PI MAX    WMAX=1500    I=00: 00: 15
```

This sample request checks every 15 seconds and sends a warning message to the BBI-SS PAS Journal log if any region exceeds 1500 PI enqueues. The region name and PST number are included in the warning message. This can be used to check if a program is taking checkpoints frequently enough.



---

## Part 2. Using MVIMS

This section describes how to use MVIMS. It includes descriptions of:

- analyzers and monitors and how they are used
- the online functions that can be used with MVIMS in a terminal session
- the Primary Option Menu, which provides easy access to product service applications

### **Chapter 3. The MAINVIEW Product Family . . . . . 39**

### **Chapter 4. Measuring IMS Activities and Resources . . . . . 41**

### **Chapter 5. Types of Services. . . . . 49**

#### **Requests . . . . . 50**

#### **Analyzers . . . . . 51**

#### **Monitors . . . . . 52**

##### **Starting and Stopping Monitors . . . . . 52**

##### **Monitor Response to Target IMS Shutdown and Startup . . . . . 52**

##### **Data Collection . . . . . 52**

##### **Historical Data Stored . . . . . 53**

##### **Data Types Measured . . . . . 54**

##### **Warning Conditions . . . . . 55**

##### **Monitor Request Title . . . . . 56**

##### **Warning Message Format . . . . . 56**

##### **IMS Monitor Data Display Services . . . . . 58**

#### **Workload Wait Events . . . . . 59**

#### **Workload Trace . . . . . 59**

#### **Logging a Display . . . . . 60**

##### **Automatic BBI-SS PAS Image Logging of Analyzer Displays . . . . . 60**

##### **Automatic BBI-SS PAS Image Logging of PLOT Display . . . . . 60**

##### **Automatic BBI-SS PAS Image Logging of Monitor Summary Displays. . . . . 61**

#### **Request Status . . . . . 62**

#### **Grouping Requests . . . . . 62**



---

## Chapter 3. The MAINVIEW Product Family

MAINVIEW is an integrated family of performance management and automation products that monitor and control the multivendor enterprise information system. MAINVIEW consists of performance monitors, network integration software, automated operations, and prewritten automation applications.

MAINVIEW product integration allows host and network system monitoring and automation (even in remote locations) through a common user interface, the MAINVIEW Selection Menu, which is shown in the *Using MAINVIEW* manual.

The integration of MAINVIEW products is provided through BMC Software Intercommunications (BBI) technology. For more information about the BBI architecture, see the *MAINVIEW Common Customization Guide*.



---

## Chapter 4. Measuring IMS Activities and Resources

The MVIMS analyzer and monitor services measure these IMS activities and resources:

- Terminal I/O
- IMS queuing
- Scheduling of application programs (PSB and DMB pools) in the dependent region
- Application program activity in the dependent regions
- Database activity and buffer pool utilization
- IMS internal functions and interactions with OS/390
- IRLM functions
- IMS workload

Table 2 on page 42 groups the analyzer and monitor services by the IMS activity or resource area measured and indicates where in this manual the services are described. The area indicated in the table is shown as it appears in the AREA field of the service list applications.

For techniques about how to use the services to tune your system, see Part 1, “Performance Analysis and Monitoring Techniques” on page 1.

Table 2. IMS Activities and Resources Measured

<b>IMS Activity/ Resource Area</b>	<b>Analyzer Displays (all references in this manual)</b>	<b>Monitors and Traces (Monitors and Traces Reference Manual)</b>
Terminal I/O (MFS area)	“MFSST - MFS Statistics” on page 83 “MFSUT - MFS Pool Utilization” on page 87	“MFSFD - Percentage of MFS Blocks Found in Pool” “MFSIO - MFS I/O” “MFSIR - MFS Immediate Requests”
IMS queuing (QUEUE area)	“QUEST - Queue Statistics” on page 89	“INQBG - Input Queue Length by Balancing Group (BALG)” “INQCL - Input Queue Length by Class” “INQTR - Input Queue Length by Transaction Code” “IQSCL - Schedulable Input Queue by Class” “LGMSG - Long Message Queue Percentage Utilization” “QBLKS - Queue Blocks Percentage of Utilization” “QIO - Queue I/O” “QWAIT - Queuing Waits” “SHMSG - Short Message Queue Percentage Utilization” “INLK - Input Messages by Link” “OQLK - Output Queue by Link” “OUTLK - Output Messages by Link” “DEADQ - Dead Letter Queue Count” “OQLN - Output Queue Length by Line” “OQLT - Output Queue Length by LTERM” “OQND - Output Queue Length by Node” “OUTLN - Messages Output by Line” “OUTLT - Messages Output by LTERM” “OUTND - Messages Output by Node”

Table 2. IMS Activities and Resources Measured (continued)

<b>IMS Activity/ Resource Area</b>	<b>Analyzer Displays (all references in this manual)</b>	<b>Monitors and Traces (Monitors and Traces Reference Manual)</b>
Scheduling of application programs in the dependent region (SCHED area)	<p>“BALGQ - BALG Queuing” on page 95</p> <p>“CLASQ - Class Queuing” on page 102</p> <p>“PSBUT, DMBUT - PSB and DMB Pool Utilization” on page 109</p> <p>“SCHED - Scheduling Statistics” on page 112</p>	<p>“ARVBG - Transactions Arrivals By Balancing Group (BALG)”</p> <p>“ARVCL - Transaction Arrivals by Class”</p> <p>“ARVPR - Transaction Arrivals By Program”</p> <p>“ARVTR - Transaction Arrivals by Transaction Code”</p> <p>“PRCBG - Transactions Processed by Balancing Group (BALG)”</p> <p>“PRCCL - Transactions Processed By Class”</p> <p>“PRCPR - Transactions Processed by Program”</p> <p>“PRCTR - Transactions Processed by Transaction Code”</p> <p>“SCHFL - Scheduling Failures by Type”</p>
Application program activity in the dependent regions (REGN area)	<p>“DLIST - DL/I Call Status” on page 118</p> <p>“DREGN - Region Detail (Event Collector Data)” on page 125</p> <p>“DREGN - Region Detail (No Event Collector Data)” on page 145</p> <p>“REGNS - IMS Regions (Event Collector Data)” on page 164</p> <p>“REGNS - IMS Regions (No Event Collector Data)” on page 187</p> <p>“STAT/STATR - System Status” on page 206</p>	<p>“D2CON - IMS Region Connection to DB2 Subsystem”</p> <p>“D2SON - DB2 Sign On by Subsystem”</p> <p>“D2THD - Active IMS Region Threads to DB2 Subsystem”</p> <p>“DBGU - DataBase Calls per Message Get Unique by Region”</p> <p>“DBTOT - DataBase Calls per Scheduling by Region”</p> <p>“MSGGU - Message Calls per Message Get Unique by Region”</p> <p>“MSGT - Message Calls per Scheduling by Region”</p> <p>“WAIT - Region in a Long PI Wait”</p>

Table 2. IMS Activities and Resources Measured (continued)

<b>IMS Activity/ Resource Area</b>	<b>Analyzer Displays (all references in this manual)</b>	<b>Monitors and Traces (Monitors and Traces Reference Manual)</b>
Database activity and buffer pool utilization (DB area)	<p>“DBST - OSAM Global Pool Statistics” on page 215</p> <p>“DBST - OSAM Subpool Statistics” on page 219</p> <p>“FPBST - Fast Path Buffer Pool Statistics” on page 223</p> <p>“VSST - VSAM Global Pool Statistics” on page 226</p> <p>“VSST - VSAM Subpool Statistics” on page 229</p>	<p>“DBIO - DataBase I/O Count by Subpool”</p> <p>“DBHIT - Hit Ratio for OSAM Buffer Pool”</p> <p>“DBSTL - DataBase Buffer Steals by Subpool”</p> <p>“HPACC - Hiperspace Access by Subpool”</p> <p>“HPHIT - Hiperspace Hit Ratio by Subpool”</p> <p>“HPSTL - Hiperspace Buffer Steals by Subpool”</p> <p>“SBUSE - Sequential Buffering Storage by Region”</p> <p>“VDBIO - VSAM Database I/O by Subpool”</p> <p>“VDBWR - VSAM Writes by Subpool”</p> <p>“VHIT - VSAM Hit Ratio by Subpool”</p>

Table 2. IMS Activities and Resources Measured (continued)

<b>IMS Activity/ Resource Area</b>	<b>Analyzer Displays (all references in this manual)</b>	<b>Monitors and Traces (Monitors and Traces Reference Manual)</b>
IMS internal functions (INTNL area)	<p>“APPCA - APPC Activity Summary” on page 233</p> <p>“APPCL - APPC LU Status” on page 242</p> <p>“DAPPC - Input Allocation Direction” on page 249</p> <p>“DAPPC - Output Allocation Direction” on page 254</p> <p>“DLTCH - Latch Detail” on page 258</p> <p>“DPOOL - Detail Pool (Non-CBT Variable Pool)” on page 268</p> <p>“DPOOL - Detail Pool (Non-CBT Fixed Pool)” on page 262</p> <p>“LATCH - Latch Summary” on page 272</p> <p>“LOGST - Log Statistics” on page 279</p> <p>“PI - Program Isolation” on page 284</p> <p>“POOLC - Pool Summary (CBT)” on page 288</p> <p>“POOLS - Pool Summary (Non-CBT Variable and Fixed Pools)” on page 291</p>	<p>“DBWP - Database Work Area Pool Percentage of Utilization”</p> <p>“DMBP - DMB Pool Percentage of Utilization”</p> <p>“DSAP - Dynamic SAP Percentage of Utilization”</p> <p>“EPCB - EPCB Pool Percentage of Utilization”</p> <p>“HIOP - HIOP Pool Percentage of Utilization”</p> <p>“LAWT - Average Latch Wait Time”</p> <p>“LMAWT - Maximum Average Latch Wait Time”</p> <p>“MFSP - MFS Pool Percentage of Utilization”</p> <p>“OBUFW - OLDS Buffer Waits”</p> <p>“OCHKW - OLDS Check Writes”</p> <p>“PIENQ - Program Isolation Enqueues by Region”</p> <p>“PIMAX - Maximum Program Isolation Enqueues by Region”</p> <p>“PIPL - Program Isolation Pool Percentage of Utilization”</p> <p>“POOLA - Pool Allocated Storage”</p> <p>“POOLN - Net Expansion Count”</p> <p>“POOLT - Total Expansion/Compression Count”</p> <p>“PSBP - PSB Pool Percentage of Utilization”</p> <p>“PSBW - PSB Work Area Pool Percentage of Utilization”</p> <p>“RECA - RECA Pool Percentage of Utilization”</p> <p>“WADIO - WADS I/O”</p> <p>“WKAP - General Work Area Pool Percentage of Utilization”</p>
IMS and OS/390 interactions (IMVS area)	<p>“DSPST - Dispatcher Statistics” on page 299</p> <p>“RS - Real Storage” on page 303</p>	<p>“CSAFR - CSA Fragmentation”</p> <p>“CSAUT - CSA Percentage of Utilization”</p> <p>“DLIO - DL/I EXCP Count by ddname”</p> <p>“DPAGE - Demand Paging by Region”</p> <p>“ECSAU - Extended CSA Percent Utilization”</p> <p>“PAGE - Paging (Region)”</p> <p>“SYSIO - EXCP Count by ddname”</p>

Table 2. IMS Activities and Resources Measured (continued)

<b>IMS Activity/ Resource Area</b>	<b>Analyzer Displays (all references in this manual)</b>	<b>Monitors and Traces (Monitors and Traces Reference Manual)</b>
IRLM functions (LOCK area)	<p>“IRLM - IRLM IMS Status (IRLM 1.5)” on page 307</p> <p>“IRLMG - IRLM GLOBAL STATUS (IRLM 1.5)” on page 313</p> <p>“LCRES - IRLM Lock Contention by Resource (IRLM 1.5)” on page 321</p> <p>“LCUSR - IRLM Lock Contention by User (IRLM 1.5)” on page 328</p> <p>“LHRES - IRLM Locks Held by Resources (IRLM 1.5)” on page 330</p> <p>“LHUSR - IRLM Locks Held by User (IRLM 1.5)” on page 337</p> <p>“LUSRD - IRLM Lock User Detail (IRLM 1.5)” on page 339</p> <p>“IRLM - IRLM IMS Status (IRLM 2.1 and Later)” on page 347</p> <p>“IRLMG - IRLM GLOBAL STATUS (IRLM 2.1 and Later)” on page 353</p> <p>“LCRES - IRLM Lock Contention by Resource (IRLM 2.1)” on page 362</p> <p>“LCUSR - IRLM Lock Contention by User (IRLM 2.1)” on page 370</p> <p>“LHRES - IRLM Locks Held by Resources (IRLM 2.1)” on page 372</p> <p>“LHUSR - IRLM Locks Held by User (IRLM 2.1)” on page 380</p> <p>“LUSRD - IRLM Lock User Detail (IRLM 2.1)” on page 383</p>	<p>“LDLCK - Number of Deadlocks”</p> <p>“LHELD - Number of Locks Held”</p> <p>“LKMAX - Maximum Locks Held by Region”</p> <p>“LKREQ - Number of Lock Requests”</p> <p>“LSUSP - Number of Suspensions”</p> <p>“LWAIT - Region in IRLM Suspend”</p> <p>“LWNUM - Number of Suspended IRLM Requests”</p> <p>“PTBLK - Number of PTB Locks”</p> <p>“VSEND - Number of VTAM Sends”</p>

Table 2. IMS Activities and Resources Measured (continued)

<b>IMS Activity/ Resource Area</b>	<b>Analyzer Displays (all references in this manual)</b>	<b>Monitors and Traces (Monitors and Traces Reference Manual)</b>
IMS Workload (IWKLD area)	<p>Chapter 18, “ISTAT - Terminal Input Status Display” on page 395</p> <p>Chapter 19, “OSTAT - Terminal Output Status Display” on page 407</p> <p>Chapter 20, “TRANQ - Transaction Queue Status Display” on page 419</p> <p>Chapter 21, “USER - User Status Summary” on page 427</p> <p>Chapter 22, “Requesting Workload Wait Data Collection (MWAIT)” on page 433</p> <p>Chapter 23, “Requesting Workload Wait Data Display (DWAIT)” on page 443</p> <p>Chapter 24, “DWAIT - Workload Wait Display” on page 445</p>	<p>Chapter 27, “Requesting Workload Trace Data Collection (MTRAC)”</p> <p>Chapter 34, “LTRAC - List of Trace Entries”</p> <p>Chapter 35, “STRAC - Summary Trace Data Display”</p> <p>Chapter 36, “DTRAC - Detail Trace Data Display”</p>
IMS workload - DB2 activity (IWDB2 area)		<p>“#CDB2 - DB2 Data Access Calls”</p> <p>“#SDB2 - DB2 Nondata Access Calls”</p> <p>“@PDB2 - DB2 CPU Time”</p>
IMS workload - Fast Path activity (IWFP area)		<p>“#CIC - Control Interval Contentions”</p> <p>“#OBAW - OBA Latch Waits”</p> <p>“@OBA - Overflow Buffer Usage”</p>
IMS workload - global IMS region calls (IWGBL area)		<p>“\$CBMP - BMP Region Calls”</p> <p>“\$CDBT - DBCTL Region DLI Calls”</p> <p>“\$CMPP - MPP Region Calls”</p> <p>“\$CTOT - All Region Calls”</p>
IMS workload transactions (IWTRN area)		<p>“@ELAP - Average Elapsed Time”</p> <p>“@INPQ - Average Input Queue Time”</p> <p>“@RESP - Average Response Time”</p> <p>“#PROC - Number of Transactions Processed”</p>



---

## Chapter 5. Types of Services

The MVIMS services include:

Analyzers	Show formatted full-screen displays of target system status and activity.
Monitors	Measure and collect data about resource or workload performance over time and detect exception conditions.
Workload Wait	Measures and collects data about IMS workload events over time.  Workload wait requires monitor data collection and analyzer display. MWAIT is a monitor service that activates workload wait data collection. DWAIT is an analyzer display of data collected by MWAIT.
Workload Trace	Tracks transaction processing through IMS.  A trace requires monitor data collection and analyzer display. MTRAC is a monitor service that activates workload trace data collection. LTRAC, STRAC, and DTRAC are analyzer display services that show different views of data collected by MTRAC from summarized statistics for all transactions to detailed events and data for one transaction.

Except when thresholds are exceeded, monitors execute independently in the background, transparent to the operator. When warnings are issued, the historical data collected by Workload Monitor and other products can be examined to determine problem causes. Workload Analyzer can be used to determine the cause of the IMS workload problem; then Resource Analyzer can be used to determine which resource is affected by the workload problem. MAINVIEW AutoOPERATOR for IMS can be used to automatically take an action when a warning is issued.

Resource Monitor has services that monitor resource usage, such as input queue length. When used together, Workload Monitor and Resource Monitor provide an early warning system for the entire IMS environment.

---

## Requests

Services are activated as user requests.

- Analyzers can be selected from an analyzer service list or requested from a service display panel by overtyping the service name.
- Timer-driven monitors that measure resources, workloads, or wait events or track transaction processing are SET service requests.

They can be selected from a monitor service list and started from a data entry panel. The collected data can be viewed with monitor display services similar to analyzers.

- Timer-controlled requests that activate transaction tracing can be started from a data entry panel accessed from a list of current traces.
- Timer-controlled requests for automatic analyzer display logging can be started from a data entry panel accessed from an analyzer service list.

SET requests for timer-driven services with options as keyword parameters can be generated automatically from data entry panels. Optional service parameters narrow the scope of a single request and let several requests for the same service be active concurrently. Additional requests can be made at any time, or requests can be modified or purged.

A standard set of timer-controlled monitor, workload wait, workload trace, or logging requests can be started automatically with the BBI-SS PAS.

---

## Analyzers

The IMS analyzers are services that provide a formatted, full-screen display of IMS resource performance at your terminal in response to a request. A request for a display is made with a service select code and optional parameter. You can make the request from a service display panel or by selecting the service from an analyzer service list. The results of the service analysis are shown at your terminal and can be automatically logged to the BBI-SS PAS Image log at time-driven intervals with the SET facility.

Analyzer services that show lists of resources or workloads are scrollable.

Some analyzers display data collected by a previously started data collection service request. For example, timer-driven monitors collect short-term history data that can be displayed with a plot display. The workload wait (MWAIT and DWAIT) and trace services (MTRAC and LTRAC, STRAC, or DTRAC) in IMS Workload Analyzer function similarly. The MWAIT and MTRAC services are started like a monitor to collect data:

- MWAIT data collection can then be displayed by the DWAIT workload wait display.
- MTRAC data collection can then be displayed by:

LTRAC	Shows a list of all traced events for an MTRAC request.
STRAC	Shows summarized trace statistics for one transaction.
DTRAC	Shows a chronological detail trace of events for one transaction and includes associated database I/O data and segment search argument, key feedback, and I/O area data. If a transaction has DB2, CICS, or MQSeries events associated with it, those events are also displayed.

---

## Monitors

The IMS monitors are services that collect data measurements about resource usage or workload performance and detect warning conditions in response to a user request. A request to activate a timer-driven monitor is made by specifying a service select code and optional parameter with the SET facility. You can make a request from a service display panel or from a monitor service list. Selecting a service from a monitor list displays a data entry panel primed with the SET request keywords for the selected service. The request specifies the resource to be measured, the sampling frequency, the time of day to begin the sampling, the sampling duration, the disposition of data when the IMS subsystem is not available, and the threshold that defines a warning condition for the monitor request.

Each monitor service is a timer-driven monitor that:

- Measures resource or workload status, count of activity, or usage percentage.
- Calculates a ratio of resource usage over time. The measurement obtained at each sampling interval is compared to the user-defined threshold for that request. The threshold comparison detects conditions for which user-defined warning messages can be issued. The measurements are stored online so that a plot of the recent history of a monitor request can be viewed at any time. Optional service parameters narrow the scope of a single request and let several requests for the same monitor service be active concurrently.

## Starting and Stopping Monitors

With a monitor request, the user selects an IMS variable to be monitored, defines a sampling frequency, and assigns a warning threshold appropriate to the site's environment. A standard set of monitor requests can be started automatically with the BBI-SS PAS. Additional monitor requests can be made at any time or requests can be modified or purged

## Monitor Response to Target IMS Shutdown and Startup

Active monitors detect target system shutdown and startup. When a request is made, quiesce and restart options can be defined that specify the monitor action when the target system stops or restarts. Monitors can:

- Be restarted automatically  
Previously collected data can be saved or deleted.
- Be purged when the target IMS stops
- Remain in a quiesced state when the target IMS starts

## Data Collection

A monitor service request is activated through a timer facility, which is controlled through SET service requests. A request for a service specifies:

- The monitor to be used (service select code and parameter).  
**Note:** This defines the IMS resource or activity to be measured.
- When monitoring should begin.

- How long monitoring should continue.
- How often the activity is to be measured (sampling interval).
- Which service functions should be performed.

The active service request automatically measures the corresponding system variable (see “Data Types Measured” on page 54) at the time interval specified on the SET request. This data can be displayed online with a plot service request. The requested plot display can be automatically logged or refreshed.

The workload monitors and resource monitors use the same timer and SET facility to activate and control requests. However, they do not use the same facility for data collection.

- Workload trace and workload monitors use the Event Collector component to collect transaction-level statistics.
- Resource monitors collect data through their own timer-driven services.

## Historical Data Stored

To make concise short- and long-term histories available for the graphic plot display, historical data is always stored as:

- Ten detail measurements. These are the latest 10 values, each collected at the expiration of the user-defined interval. For example, if the standard sampling interval of 1 minute is in effect, the measurements of each of the last 10 minutes are available. When a new measurement is made, the oldest value is overwritten (in other words, the values wrap around every 10 intervals).
- Two summary periods, current and previous. Both values are updated at the expiration of 10 intervals (wrap point) when 10 new detail measurements have been collected. The current period value is moved to the previous period and the sum of the 10 detail measurements is moved to the current period. The current period value includes from 1 to 10 of the detail measurements available at any one time. The wrap point is indicated by an arrow in the display provided by the PLOT service.
- Total. This is the total accumulated in the time the request has been active. It is updated at each interval when a new measurement is made.

**Note:** Both the summary periods and the total are shown as averages-per-interval in the graphic display so the plotted detail values can be compared.

- A frequency distribution. This distribution is updated at each interval if range limits are defined with the request. From two to five ranges are allowed. The new measurement value is compared to the defined limits to find the range in which it belongs and the number of occurrences for that range is incremented by one.
- The high-water mark. This is the maximum value ever measured at any interval and the time it occurred.

Following is an example of the storage of historical data:

If a request is started at 10:00 a.m. with an interval of one minute, the detail counters wrap around at 10:10, 10:20, and so on. The history available at 10:35 is:

- The detail measurements cover the last 10 minutes, from 10:25 to 10:35.
- The current period is from 10:20 to 10:30.
- The previous period is from 10:10 to 10:20.
- The total is from 10:00 to 10:35.

## Data Types Measured

The following four types of automatic service measurements can be taken periodically and shown by the general Performance Management PLOT display request.

### COUNT

An activity count over time; for example, the number of lock requests within a specified time interval.

**Note:** When COUNT data is plotted, in addition to the counts, rates-per-second are automatically calculated and shown for the displayed time intervals (AVG/SEC).

### AVERAGE

The quantity over time; for example, the average value of transaction response time in the specified interval.

**Note:** When AVERAGE data is plotted, in addition to the averages, the event counts used to calculate the averages are also shown for the displayed time intervals (EVENTS); for example, for the plotted average response time, the number of transactions measured is shown.

### STATUS

The status level at the moment of measurement; for example, queue lengths or level of PI enqueues.

### PERCENT

Resource usage at the moment of measurement, expressed as a percentage of the maximum; for example, percent pool utilization.

For a sample PLOT display and a complete description of the display contents, see the “Monitor History Display (PLOT)” section of the “Monitor Display Commands” chapter in *Monitors and Traces Reference Manual*.

A fifth type of data measurement, WARNING ONLY, does not collect historical data or produce a plot:

### WARNING ONLY

A condition measurement that can be checked against a warning threshold; for example, the number of DL/I calls performed in a region since the last program scheduling.

## Warning Conditions

SET request parameters for a monitor service can be used to define a value that is compared to the measurement taken during the requested sampling interval. The comparison establishes a warning condition when the measurement either exceeds a maximum threshold or is less than a minimum threshold. When the service detects this exception, it automatically sends a message to the BBI-SS PAS Journal and also to the system console upon user request. The message text is:

- A unique message ID
- The title of the service, which can be changed by the user with a TITLE parameter in the service request
- The measured value
- The sampling interval (if applicable)
- The defined threshold

The service sends an exception-cleared message with the service title to the BBI-SS PAS Journal when the service no longer detects a condition greater than the defined threshold value.

The monitor request can be used to:

- Specify a user-defined threshold value (WMAX or WVAL keyword).

Each new measurement of the system variable made at the expiration of an interval is compared to the threshold value.

When the measurement either exceeds a maximum threshold or is less than a minimum threshold, a warning condition exists and warning messages are sent automatically to the BBI-SS PAS Journal log.

**Note:** The LOG DISPLAY option on the Primary Option Menu can be used to view the BBI-SS PAS Journal log.

- Send warning messages also to the OS/390 console through the write-to-operator (WTO) facility (WMSG keyword).
- Specify the number of warnings to be sent for one exception condition (WLIM keyword), the number of times the exception is detected before the first message (WIF keyword), and the number of times the exception is detected between messages (WIN keyword).

These options can be used to avoid flip-flop situations where a condition often varies just above and below the threshold, triggering many messages. For example, a condition could be checked every 30 seconds with a warning only if that condition persists for 3 minutes (WIF=6), repeated warnings only after another 5 minutes (WIN=10), and a limit of 10 (WLIM=10) warnings (the problem is known and investigated by then).

## Monitor Request Title

A monitor request is identified with a title. If a parameter is specified for a requested service, the parameter is shown with the title. The title and applicable parameter for each request are shown in the:

- PLOT graphic display of the data collected by the requested monitor
- Active Timer Requests application (see “Request Status” on page 62)
- DMON or DWARN active monitor summary display
- Warning message issued when the measurement of the resource exceeds a threshold defined for the monitor

If a parameter is not used for the monitor request, the PLOT display and the warning message show the default (TOTAL) for that field.

Each service has a default title that can be customized. This process is described in “Service Table Definition” in the *MVIMS Customization Guide*. The default titles of all the monitor services are in the service descriptions in this manual.

When requesting a monitor service, you can use the TITLE keyword to make the title more meaningful to the installation for that specific request. A user-defined title can be 1 to 24 characters long. The title can be defined in a data entry panel or with a SET request in the Service Display panel. If a title is defined with a SET request, the title must be enclosed in single quotes.

## Warning Message Format

Each service has a unique warning message associated with it. A warning message is issued when the condition established by the user is detected by the monitor service, as described in the preceding section. The format of a warning message is:

```
ccnnn0W (nn) hh:mm:ss title(parm) = v [IN x intrvl] [srvdata]  
(>thrshl d) *****
```

where:

ccnnn0W	Is the warning message ID issued by the requested monitor service.
cc	Is a two-character code indicating the service type; for example, RM indicates Resource Monitor.
nnn	Is the numerical message identifier associated with the requested service.
0	Indicates that the detected threshold currently exists.
W	Represents a warning message.
(nn)	Is the number of times the warning message was issued.
hh:mm:ss	Is a timestamp in hours, minutes, and seconds.
title	Is a default or user-defined title for the service (see “Monitor Request Title” on page 56).

(parm)	Is an optional parameter that is part of the reqid (service select code plus parameter) for the SET request, as described previously. (TOTAL) is the default if a parameter is not specified for the service request.				
v	Is the current measured value.				
IN x intrvl	Is the time specified for the resource sampling with the INTERVAL keyword of the SET request where: <table> <tr> <td>x</td><td>Can be nn, nnnn, hh:mm:ss, or mm:ss (n is a numeric value; hh is the number of hours; mm is the number of minutes; and ss is the number of seconds).</td></tr> <tr> <td>intrvl</td><td>Is units of time measurement which can be SEC or MIN.</td></tr> </table> <p><b>Note:</b> Time measurement units are not used for hh:mm:ss.</p> <p>This measurement is included in the warning message only when a COUNT data type is measured (see “Data Types Measured” on page 54).</p>	x	Can be nn, nnnn, hh:mm:ss, or mm:ss (n is a numeric value; hh is the number of hours; mm is the number of minutes; and ss is the number of seconds).	intrvl	Is units of time measurement which can be SEC or MIN.
x	Can be nn, nnnn, hh:mm:ss, or mm:ss (n is a numeric value; hh is the number of hours; mm is the number of minutes; and ss is the number of seconds).				
intrvl	Is units of time measurement which can be SEC or MIN.				
(>thrshld)	Is the threshold value specified by the WMAX or WVAL keyword of the SET request for the monitor service. A < character indicates the sampled value is less than or equal to the threshold as specified by the request.				
*****	Is used to emphasize the message in the BBI-SS PAS Journal log.				

When a threshold is exceeded, a warning message is sent and a warning condition exists. The DWARN service can be used to show all current warning conditions. When the condition that caused the warning no longer exists, the following message is issued:

```
ccnnn1I hh:mm:ss title(parm) NO LONGER > value
```

where:

ccnnn1I	Is the same as the warning message ID number except a 1 replaces the 0 in the last digit and I replaces W. The 1 indicates the detected threshold no longer exists. The I indicates this message is informational.
title	Is the same as the warning message title.
(parm)	Is the same as the warning message parameter.
value	Is the threshold value specified by the WVAL keyword of the SET request for the monitor service.

For example, if the user request is:

```
LKREQ - IRLM LOCK REQUESTS
```

```
WVAL      ==> 5
WMSG      ==> WTO
INTERVAL  ==> 00:01:00
TITLE     ==> IRLM LOCK REQUESTS
```

and the sampled measurement is greater than 5 threads at 1:00 pm, the following RM0840W message is issued:

```
RM0840W(01) 13:00:01 IRLM LOCK REQUESTS(TOTAL) = 7 (>5)
```

When the condition no longer exists, the following RM0841I message is issued:

```
RM0841I (01) 13:31:00 IRLM LOCK REQUESTS(TOTAL) NO LONGER > 5
```

**Note:** The target system is identified in these messages. In the Journal log, the target name is in the origin identifier field (scroll left to view). WTO messages have both the BBI-SS PAS ID and the target (TGT=xxxx) appended at the end of the message text.

## IMS Monitor Data Display Services

The data collected by monitor service requests can be displayed online or the displays can be logged for later analysis. They can be requested from an active timer request list (see the chapter “Displaying a List of Active Timer Requests” in *Monitors and Traces Reference Manual*).

The monitor-collected data can be displayed by the following services:

- PLOT

The PLOT service provides a graphic display of the history data collected for one monitor request. A range distribution of the measured values, the maximum value ever measured, and rates-per-second present effective IMS problem analysis.

- DMON

The DMON service provides a scrollable display of the most current, active monitor measurements. Each line has a simple graphic representation of how close the measurement is to the defined warning threshold.

DMON identifies potential problems quickly by showing several measurements together, such as the number of transactions processed, the number of database I/Os, the level of PI enqueues, and average response time. You also can use this service to see how many monitors are close to their warning thresholds.

- DWARN

The DWARN service provides a scrollable display of current, measured values like DMON, but only for monitors that have a warning condition resulting from measured values exceeding user-defined thresholds.

These displays are described in detail in the chapter called “Monitor Display Commands” in the MAINVIEW for IMS *Online – Monitors and Traces Reference Manual*.

---

## Workload Wait Events

The workload wait services collect and display workload wait events for all or specific IMS workloads.

As described in Chapter 22, “Requesting Workload Wait Data Collection (MWAIT)” on page 433, workload wait events are sampled by a request for the MWAIT monitor. The request can be tailored to select specific components of the IMS workload for accumulated wait time. For example, the only workload components that use an IMS region are scheduling, application program, and sync point. Input and output communication and queuing events are ignored when a REGION parameter is specified.

The wait data accumulated by MWAIT is viewed by a request for the DWAIT workload analyzer. DWAIT shows the workload events that account for the IMS response time. As described in Chapter 24, “DWAIT - Workload Wait Display” on page 445, DWAIT shows wait events by the following transaction processing event components:

- Input Communications
- Input Queue
- Scheduling
- Application Program
- Sync Point
- Output Queue
- Output Communications

All or specific transaction processing event components can be viewed.

---

## Workload Trace

The trace services collect and display trace data about transaction processing and allow traced data to be logged to external VSAM data sets called trace log data sets (TLDS). Active traces can be viewed online with the Current Traces application from the Primary Option Menu. Logged traces can be viewed online with the History Traces application from the Primary Option Menu.

The Event Collector must be active to implement a trace. As described in the MTRAC chapter in *Monitors and Traces Reference Manual*, a trace is implemented by a request for the MTRAC monitor. The request can be tailored so that only the trace data needed to detect and solve problems is collected. The request can specify either a summary or detail trace.

The traced data collected by an MTRAC request can be viewed by requesting the LTRAC, STRAC, or DTRAC services. Display of a summary trace (LTRAC or STRAC) provides high-level quick answers about a transaction as it flows through IMS. Display of a detail trace (DTRAC) provides a chronological replay of the exact sequence of traced transaction events and includes associated database I/O data and segment search argument, key feedback, and I/O area data. If a transaction has DB2, CICS, or MQSeries events associated with it, those events are also displayed.

---

## Logging a Display

A display can be logged in three different data sets by:

- Entering a Y for the yes option in the LOG field of the display, which records the display in the TS Image log.
- Pressing the SCREEN PRINT key (PF4/16) after the display is shown, which records the display to a BBISPRNT data set.
- Requesting automatic logging of an analyzer or monitor display service, which records the display automatically, without user interaction, to the BBI-SS PAS Image log.

DMON and DWARN display logging can also be requested for active monitors as described in “Automatic BBI-SS PAS Image Logging of Monitor Summary Displays” on page 61.

The log records can be printed offline using the IMRPRINT utility (BBSAMP member ILOGJCL), as described in the *MAINVIEW Administration Guide*.

## Automatic BBI-SS PAS Image Logging of Analyzer Displays

An Image log request can be made by selecting a data entry panel from an analyzer service list with the I line command. The SET timer facility invokes an IMS analyzer service and logs the display automatically to the BBI-SS PAS Image log at a user-specified interval. ATI NTVL is the default. For example:

```
SERV ==> SET  
PARM ==> REQ=PI, I=00: 05: 00
```

requests Image logging of the program isolation display every five minutes. For a description of the keywords used to activate an Image log request either with a SET request (see the “SET Timer Request” chapter in *Monitors and Traces Reference Manual*) or from the Image log request data entry panel (see “Start Image Log Request (I Line Command)” on page 71).

## Automatic BBI-SS PAS Image Logging of PLOT Display

Logging of a monitor PLOT display to the BBI-SS PAS Image log is coordinated automatically by specifying a LOG parameter with a monitor request as described in the chapter “Requesting a Monitor” in *Monitors and Traces Reference Manual* or with a SET request as follows:

```
SERV ==>SET  
PARM ==>REQ=HPSTL, 1, I=00: 01: 00, LOG=ATPD
```

This request invokes the HPSTL data collection monitor to collect the number of unsuccessful Hiperspace reads for VSAM subpool 1 at one minute intervals. At the end of each complete period (LOG=ATPD), which is 10 intervals, a plot display of the data is logged to the BBI-SS PAS Image log. For this request, a plot is logged every 10 minutes.

A convenient logging frequency for a complete monitor history is at the end of each period (ATPD). A period is the completion of 10 time intervals. A PLOT display can also be logged at each interval (LOG=ATI NTVL), only once at the completion of the request (LOG=ATSTOP), or only when a warning condition is detected (LOG=ATWARN).

## Automatic BBI-SS PAS Image Logging of Monitor Summary Displays

A summary of active monitor status can be logged to the Image log with DMON or DWARN, as shown by the following requests which can be made from the SERV field of any display.

For example, this request logs the DMON service display, which shows the current status of the first 15 active monitors, every 10 minutes:

```
SERV ==> SET
PARM ==> REQ=DMON, I=00: 10: 00
```

The following request logs the DWARN service display, which shows the current status of the first 15 active monitors with a warning condition, every minute:

```
SERV ==> SET
PARM ==> REQ=DWARN, I=00: 01: 00
```

The table called “SET Keywords Affecting Request Activation” in *Monitors and Traces Reference Manual* describes the keywords used to request Image logging of a DMON or DWARN display.

**Note:** If a parameter is not specified for a DMON or DWARN Image log request, the first 15 active monitors are logged. To log the next set of 15 active monitors, specify 16 in the PARM field, and so on.

---

## Request Status

The status of timer-driven requests can be displayed by accessing the Active Timer Requests application:

Select Primary Menu Option 2, MONITORS - Early Warnings/Recent History (Active Timer Requests)

The Active Timer Requests application lists standard monitor service requests and any additional timer-driven data collection services and timer-driven Image logging requests. It shows how many requests are active already and provides direct access to the data collected by that request. You can access data entry panels that allow current options to be viewed or modified, purge an active request, or use a request as a model to start a new request.

The Display Statistics and Defaults application, Primary Menu Option 5 from the Primary Option Menu, provides general information about the Timer Facility. It shows status information, some statistics, default parameters in effect, and a summary of the active timer requests for the BBI-SS PAS associated with the specified target (TGT==>).

---

## Grouping Requests

Multiple timer-driven services can be started by defining a series of requests in a member of the BBI-SS PAS BBPARM data set. The member can be started from a TS or automatically when the BBI-SS PAS starts, as described in the “Request Initiation” section of the “SET Timer Request” chapter in *Monitors and Traces Reference Manual*.

BBPARM member, BLKIMFW, contains a sample starter set of IMS Resource Monitor and IMS Workload Monitor requests. Many of the requests have suggested warning thresholds; some only show activity in the IMS target.

This member should be used for automatic monitor startup (see BBPARM member BBIISP00) until a set of monitors can be customized for each IMS target at your site.

---

## Part 3. Analyzers

This section summarizes how to request a service and describes what each service does. The service descriptions are organized into groups that parallel the transaction processing sequence within IMS.

<b>Chapter 6. Displaying a List of Analyzer Services (Menu Option 1)</b>	67
SORT Primary Command	68
AREA Primary Command	69
Line Commands	69
Start Image Log Request (I Line Command)	71
<b>Chapter 7. Requesting an Analyzer Display</b>	73
Selecting a Service(s) for Display	73
Logging an Analyzer Service Display	74
Stopping an Analyzer Image Log Request	75
Qualifying an Analyzer Request	75
<b>Chapter 8. Resource Analyzer Services (Quick Reference)</b>	77
<b>Chapter 9. Message Format Service Displays</b>	83
MFSST - MFS Statistics	83
MFSUT - MFS Pool Utilization	87
<b>Chapter 10. Queuing Displays</b>	89
QUEST - Queue Statistics	89
<b>Chapter 11. Scheduling Displays</b>	95
BALGQ - BALG Queuing	95
CLASQ - Class Queuing	102
PSBUT, DMBUT - PSB and DMB Pool Utilization	109
SCHED - Scheduling Statistics	112
<b>Chapter 12. Region Displays</b>	115
DLIST - DL/I Call Status	118
DREGN - Region Detail (Event Collector Data)	125
DREGN - Region Detail (No Event Collector Data)	145
REGNS - IMS Regions (Event Collector Data)	164
REGNS - IMS Regions (No Event Collector Data)	187
STAT/STATR - System Status	206
<b>Chapter 13. Database Displays</b>	215
DBST - OSAM Global Pool Statistics	215
DBST - OSAM Subpool Statistics	219
FPBST - Fast Path Buffer Pool Statistics	223
VSST - VSAM Global Pool Statistics	226
VSST - VSAM Subpool Statistics	229
<b>Chapter 14. IMS Internals Displays</b>	233
APPCA - APPC Activity Summary	233
APPCL - APPC LU Status	242
DAPPC - Input Allocation Direction	249
DAPPC - Output Allocation Direction	254

DLTCH - Latch Detail .....	258
DPOOL - Detail Pool (Non-CBT Fixed Pool) .....	262
DPOOL - Detail Pool (Non-CBT Variable Pool) .....	268
LATCH - Latch Summary .....	272
LOGST - Log Statistics .....	279
PI - Program Isolation .....	284
POOLC - Pool Summary (CBT) .....	288
POOLS - Pool Summary (Non-CBT Variable and Fixed Pools) .....	291
 <b>Chapter 15. OS/390 System Displays</b> .....	 299
DSPST - Dispatcher Statistics .....	299
RS - Real Storage .....	303
 <b>Chapter 16. IRLM Displays</b> .....	 307
IRLM - IRLM IMS Status (IRLM 1.5) .....	307
IRLMG - IRLM GLOBAL STATUS (IRLM 1.5) .....	313
LCRES - IRLM Lock Contention by Resource (IRLM 1.5) .....	321
LCUSR - IRLM Lock Contention by User (IRLM 1.5) .....	328
LHRES - IRLM Locks Held by Resources (IRLM 1.5) .....	330
LHUSR - IRLM Locks Held by User (IRLM 1.5) .....	337
LUSRD - IRLM Lock User Detail (IRLM 1.5) .....	339
IRLM - IRLM IMS Status (IRLM 2.1 and Later) .....	347
IRLMG - IRLM GLOBAL STATUS (IRLM 2.1 and Later) .....	353
LCRES - IRLM Lock Contention by Resource (IRLM 2.1) .....	362
LCUSR - IRLM Lock Contention by User (IRLM 2.1) .....	370
LHRES - IRLM Locks Held by Resources (IRLM 2.1) .....	372
LHUSR - IRLM Locks Held by User (IRLM 2.1) .....	380
LUSRD - IRLM Lock User Detail (IRLM 2.1) .....	383
 <b>Chapter 17. WORKLOAD ANALYZER Displays (Quick Reference)</b> .....	 393
 <b>Chapter 18. ISTAT - Terminal Input Status Display</b> .....	 395
Input LTERM Status .....	398
 <b>Chapter 19. OSTAT - Terminal Output Status Display</b> .....	 407
Output LTERM Status .....	412
 <b>Chapter 20. TRANQ - Transaction Queue Status Display</b> .....	 419
Transaction Status .....	423
 <b>Chapter 21. USER - User Status Summary</b> .....	 427
 <b>Chapter 22. Requesting Workload Wait Data Collection (MWAIT)</b> .....	 433
Starting MWAIT .....	433
Using the Workload Wait Data Entry Panel .....	435
Stopping MWAIT .....	440
Storage Requirements .....	440
Defining a Sampling Interval .....	441
Qualifying a Workload MWAIT Request .....	441
 <b>Chapter 23. Requesting Workload Wait Data Display (DWAIT)</b> .....	 443

<b>Chapter 24. DWAIT - Workload Wait Display</b> .....	445
Excluding Queued Events .....	450
DWAIT Event Definitions .....	450
Application Program (AP) Events .....	451
Input Communications (IC) Events .....	468
Input Queue (IQ) Events .....	477
Output Communications (OC) Events .....	480
Output Queue (OQ) Events .....	481
Scheduling (SC) Events .....	484
Synchronization Point (SP) Events .....	498



## Chapter 6. Displaying a List of Analyzer Services (Menu Option 1)

This application is a scrollable list of all the analyzer display services that you are authorized to view.

BMC SOFTWARE		----- ANALYZER DISPLAY SERVICES -----			PERFORMANCE MGMT	
COMMAND ==>					TGT ==> IMSA	
COMMANDS: SORT, AREA						
LC CMDS: S(SELECT), I(IMAGE LOGGING), H(HELP)						
LC	SERV	PARAMETER	TITLE	PARM TYPE	SEC	AREA STAT
	DWAIT		DISPLAY WORKLOAD WAIT	(IDENTIFIER)	A	IWKLD
	DTRAC		DISPLAY WORKLOAD TRACE	(IDENTIFIER)	A	IWKLD
	TRANQ		TRANSACTION QUEUE STATUS	(IDENTIFIER)	A	IWKLD
	ISTAT		TERMINAL INPUT STATUS	(IDENTIFIER)	A	IWKLD
	OSTAT		TERMINAL OUTPUT STATUS	(IDENTIFIER)	A	IWKLD
	MFSST		MFS STATISTICS		A	MFS
	MFSUT		MFS POOL UTILIZATION		A	MFS
	QUEST		QUEUE STATISTICS		A	QUEUE
	CLASQ		CLASS QUEUING	(REGION#)	A	SCHED
	BALGQ		BALG QUEUING	(REGION#)	A	SCHED
	SCHED		SCHEDULING STATISTICS		A	SCHED
	PSBUT		PSB POOL UTILIZATION	(ITERATION#)	A	SCHED
	DMBUT		DMB POOL UTILIZATION	(ITERATION#)	A	SCHED
	REGNS		IMS REGIONS	(IDENTIFIER)	A	REGN
	DREGN		DETAIL REGION SERVICE	(REGION#)	A	REGN
	REGND		DETAIL REGION SERVICE	(REGION#)	A	REGN

Figure 1. List Analyzer Display Services Application

It allows service selection by line command and shows the allowable parameters for each service, the service security classification, the area of IMS being analyzed, and the service status by:

Field Name	Description
LC	A line command input field. One-character line commands can be entered in this field to execute a service and display the output, display a data entry panel to define and submit a SET timer request for BBI-SS PAS Image logging, or display HELP information for the selected service (see “Line Commands” on page 69).
SERV	A scrollable list of all the analyzer services by service select code.
PARAMETER	An input parameter field.
TITLE	The service title.
PARM TYPE	A short description of the parameters that can be used, if the service allows parameters. Optional parameters are shown in parentheses.
SEC	The security code for user access to the service.

AREA	The IMS resource area being analyzed. This field could contain:	
	<b>Field Data</b>	<b>Description</b>
	MFS	Terminal I/O
	QUEUE	IMS queuing
	SCHED	Scheduling of application programs in the dependent region
	REGN	Application program activity in the dependent regions
	DB	Database activity and buffer pool utilization
	INTNL	IMS internal functions
	IMVS	IMS and OS/390 interactions
	LOCK	IRLM functions
	IWKLD	IMS workload
STAT	The service status (LOCK or blank).	

---

## SORT Primary Command

When the list of analyzer display services is displayed initially, the list is sorted by an internally defined sequence. SORT can be entered in the COMMAND field of the display to sort the list by any of the following column headings. The first two characters of the column heading are used with SORT as follows:

**SORT cc**

where cc can be any of the following two characters:

SE	Sorts the list alphabetically by service name (SERV column).
TI	Sorts the list alphabetically by service title (TITLE column).
SC	Sorts the list alphabetically by security code (SEC column).
AR	Sorts the list alphabetically by the resource area (AREA column) and by service name within the area (default).
ST	Sorts the list alphabetically by the service status displayed (STATUS column).

SORT with no parameters sorts the columns by area.

---

## AREA Primary Command

You can use the AREA command to list only the services related to a specific area. The possible areas that can be specified are shown in the AREA column. For example, to list only the IMS database services, type in the COMMAND field:

AREA DB

Type AREA without an area name to return to the list of all the services.

---

## Line Commands

Entering one of the following one-character line commands in the LC field for a service executes the line command function. Multiple line commands can be entered at one time and are processed in sequence. Each display in a series is shown by pressing the END key. Each Image log in a series is submitted by pressing the ENTER key and then the END key to process the next request.

Line Command	Description
S	SELECT. Selects a resource or workload analyzer service for execution and displays the output of the service analysis in the service display panel.

### Service Display Panel Input Fields

Additional service requests can be made in the following input fields of the selected service display panel:

Input Field	Description
SERV	The service select code of the selected analyzer is shown in this field. Any of the analyzer service select codes can be entered in this field. Entering a valid service select code requests the specified service and shows the results of its analysis in the display panel's output data lines (see the "Output Fields" description below). Pressing the END key redisplay the Analyzer Display Services list.
PARM	An optional parameter (maximum of 55 characters).
INTVL	The screen refresh interval (3 second default).
LOG	Writes the screen image to the Terminal Session Image log (Y N).
TGT	The target IMS job name or IMSID.

SCROLL	The scroll amount for lists of data, which can be from the cursor position (CSR), a specified number of lines, half a page (H), a page (P), or the maximum list elements (M). The scroll direction is determined by the use of the UP (PF7/19) or DOWN (PF8/20) key.
--------	--

### Service Display Panel Output Fields:

The output fields of the selected service display panel provide the following information:

Output Field	Description
INPUT RUNNING	INPUT indicates the screen is in input mode. RUNNING indicates the screen is in refresh mode. The GO key (PF6/18) changes input mode to refresh mode. Pressing the ATTN key (SNA terminals) or the PA1 key (non-SNA terminals) changes refresh mode to input mode. For information about the service refresh cycle option, see the <i>Using MAINVIEW</i> manual. refresh.
hh:mm:ss	Timestamp.
Service Title	The service description.
data lines	The service analysis output. List data can be scrolled when it fills the screen.

Pressing the END key (PF3/15) redisplay the Analyzer Display Services list.

I	IMAGE LOGGING. Displays the data entry timer request panel to specify when the service is to be invoked and the output is to be automatically recorded in the BBI-SS PAS Image Log (see “Start Image Log Request (I Line Command)” on page 71). A display is logged at the end of each interval.
H	HELP Displays HELP information for this service. This shows the service title, gives a short description, defines any parameters, and describes all fields.

The following commands are for system programmer use and are restricted by a security access code:

L	LOCK. Locks this service. The service cannot be used again until it is unlocked.
U	UNLOCK. Unlocks this service, which could be locked by the use of the LOCK command or a service ABEND.
T	TEST. BMC Software use only.

# Start Image Log Request (I Line Command)

This panel shows the options that can be specified to automatically invoke the selected analyzer service and log the output to the BBI-SS PAS Image log. The input fields are prefixed with a highlighted ==> symbol. Any default values for a field are displayed.

BMC SOFTWARE	-----	START IMAGE LOGGING REQUEST	-----	PERFORMANCE MG
COMMAND ==>				TGT ==> IMSA
RGNPI - REGION PROGRAM ISOLATION				
PARM	==>		(REGION#)	
INTERVAL	==>	00: 01: 00	START ==>	STOP ==> QIS ==> YES
RST	==>	HOT	(Restart Option: HOT, COLD, PUR, QIS)	

Figure 2. Image Log Request Application

Each request must be unique and is defined by the service select code and an optional parameter (reqid). The parameter is required if the same service is requested more than once. The service field is preset with the code of the selected service.

You can enter:

- INTERVAL

Time interval when image logging of this service is to automatically occur.
- QIS

What the service is to do when IMS is not active:

YES

Quiesce.

NO

Continue running or start.
- RST

How service is to restart after quiesce:

COLD

Restart automatically, deleting collected data.

HOT

Restart automatically without data loss.

PUR

Purge automatically when IMS starts.

QIS

Remain quiesced until purged by authorized user.
- START

The time image logging is to start in hh: mm: ss.
- STOP

The time image logging is to stop in hh: mm: ss.

For more information about these keywords, see the “SET Timer Request” chapter in *Monitors and Traces Reference Manual*.

The request is submitted when the ENTER key is pressed. A short message in the upper-right corner of the display shows the result of the request. If an ERROR IN REQUEST message is displayed, a short explanatory message is displayed on the third line. Pressing the END key (PF3/15) redisplay the Analyzer Display Services list.



---

## Chapter 7. Requesting an Analyzer Display

You can issue requests to:

- Access the analyzer displays easily through ISPF-like menus and scrollable lists
- Move quickly from summary displays to detail displays or to related services (EXPAND)
- View displays that are refreshed in a user-defined cycle
- Invoke an analyzer display from an MAINVIEW AutoOPERATOR EXEC
- Invoke a new display from the current display
- Print a screen image automatically to the online BBI-SS PAS Image log, the TS Image log, or your BBISPRNT data set

These display request and logging methods are described in the following sections.

---

### Selecting a Service(s) for Display

You can request an analyzer display by:

- Selecting one or more services from an analyzer service list

Use the S line command from the Analyzer Display Services list application to select a service (see Chapter 6, “Displaying a List of Analyzer Services (Menu Option 1)” on page 67).

- EXPANDing from another display

For displays with <<EXPAND>>, move the cursor to a row or column in the display and press ENTER to select:

- A related analyzer display
- More detailed information about a data row in the display

For displays with an EXPAND: selection line, move the cursor to a field in the selection line and press ENTER to select:

- A related analyzer display
- More detailed information for the first data row in the display

- Setting up displays for timed, cyclic refresh

Select Option C, CYCLE SETUP, from the Primary Option Menu to set up a continuous timed cycle of refreshable displays (see the *Using MAINVIEW* manual for more information).

- Invoking an analyzer display from an MAINVIEW AutoOPERATOR EXEC

Write an EXEC that invokes a resource analyzer service (MAINVIEW AutoOPERATOR must be installed). Use the IMFEXEC IMFC command followed by the service name, optional parameters, and an identifier for the target system; for example:

```
IMFEXEC IMFC TRANQ TARGET=IMS
```

The display data is returned in the predefined &LINE04 - &LINE43 variables to assist in making informed automation decisions.

- Invoking an analyzer display from a service display

When viewing any display, you can overtype the values in the SERV and PARM fields to select any other display.

---

## Logging an Analyzer Service Display

Service displays can be written to a data set for later viewing by:

- Logging a screen image to your BBISPRNT data set  
Press the PF4/16 key after the display is presented.
- Logging a display image record to your TS Image log  
Enter a Y in the LOG field of the display.
- Logging a display image record automatically at timer-driven intervals to your BBI-SS PAS Image log

Use the:

- I line command in the list of Analyzer Display Services as described in Chapter , “Start Image Log Request (I Line Command)” on page 71

BBSAMP member SLOGJCL can be used to create a hardcopy of BBISPRNT. BBSAMP member ILOGJCL can be used to create a hardcopy of the Image log records.

**Note:** BBISPRNT contains only screen images, while the Image log records contain the complete data produced by the display request, even when the data has been produced by scrolling through multiple screens.

---

## Stopping an Analyzer Image Log Request

BBI-SS PAS timer-driven Image log requests can be stopped by:

- Purging the request from the Active Timer list application

Use the P line command from the Active Timer list, as described in the chapter called “Displaying a List of Active Timer Requests” in *Monitors and Traces Reference Manual*.

- Using a SET request as described in *Monitors and Traces Reference Manual*
  - Purge the request

Issue a purge (PRG) with a SET request from the Service Display panel, BBPARM (see “Grouping Requests” on page 62), or an MAINVIEW AutoOPERATOR EXEC IMFC command (MAINVIEW AutoOPERATOR must be installed); for example:

```
SET
PRG=requ d|ALL
```

- Stop the request automatically

Use the STOP or STOPCNT parameter with the SET request from the Service Display panel or an MAINVIEW AutoOPERATOR EXEC; for example:

```
SET
REQ=TRANQ, I=00: 05: 00, START=24: 00: 00, STOP=06: 00: 00
```

---

## Qualifying an Analyzer Request

Requests for multiple transactions or terminals with similar names can be made by using a + or \* character as a name qualifier. The \* character is used generically and the + character is used positionally. As a generic resource name qualifier, the \* cannot be followed by any other character. As a positional qualifier, the + must be repeated for every character to be replaced.

For example, selecting TRANQ (Primary Option Menu 1) and entering the following in the PARM field:

```
PARM ==> START=PAY*, NONZERO
```

Is a request to display transaction codes with one or more messages queued for processing and to start with transaction codes that begin with PAY.

Entering the following in a service display panel:

```
SERV ==> TRANQ
PARM ==> TRAN=PA++T
```

Shows the transaction queue status of all transaction codes that begin with PA and end with T.

Selecting OSTAT (Primary Option Menu 1) and entering the following in the PARM field of the display:

PARM ==> LTERM=\*, NONZERO

Shows all LTERMs with queued messages.

Qualifiers can be used when making workload status display requests with the following parameters:

<b>Parameter</b>	<b>Status Display Service</b>
LTERM	ISTAT, OSTAT
TRAN	TRANQ

## Chapter 8. Resource Analyzer Services (Quick Reference)

The following table is an alphabetical list of all the Resource Analyzer services and their parameters, with page references to a more detailed description about their use. See Chapter 6, “Displaying a List of Analyzer Services (Menu Option 1)” on page 67 for a complete description of the analyzer display services.

Table 3. Resource Analyzer Service Select Codes

Service Select Code	Parameter	See
APPCA	[,CLS=nnn ,CONVID=hex ,D=I   0 ,LU=name ,Q>n ,RGN=nnn ,TR=trancode ,US=i d ,XON ,S0=cc ]	“APPCA - APPC Activity Summary” on page 233
APPCL	[,D=I   0 ,LU=name ,Q>n ,XON ,S0=cc ]	“APPCL - APPC LU Status” on page 242
BALGQ	[regi onnum]	“BALGQ - BALG Queuing” on page 95
CLASQ	[regi onnum]	“CLASQ - Class Queuing” on page 102
DAPPC	[,CONVID=hex]	“DAPPC - Input Allocation Direction” on page 249  “DAPPC - Output Allocation Direction” on page 254
DBST		“DBST - OSAM Global Pool Statistics” on page 215
DBST	subpool num	“DBST - OSAM Subpool Statistics” on page 219
DLIST	[regi onnum]	“DLIST - DL/I Call Status” on page 118
DLTCH	[genl tch- i d, H   W]	“DLTCH - Latch Detail” on page 258
DMBUT	[i teratnnum]	“PSBUT, DMBUT - PSB and DMB Pool Utilization” on page 109
DPOOL	{pool name}	“DPOOL - Detail Pool (Non-CBT Fixed Pool)” on page 262  “DPOOL - Detail Pool (Non-CBT Variable Pool)” on page 268

Table 3. Resource Analyzer Service Select Codes (continued)

Service Select Code	Parameter	See
DREGN	[regionnum]	“DREGN - Region Detail (Event Collector Data)” on page 125  “DREGN - Region Detail (No Event Collector Data)” on page 145
DSPST	[regionnum]	“DSPST - Dispatcher Statistics” on page 299
FPBST		“FPBST - Fast Path Buffer Pool Statistics” on page 223
IRLM		“IRLM - IRLM IMS Status (IRLM 1.5)” on page 307
IRLM		“IRLM - IRLM IMS Status (IRLM 2.1 and Later)” on page 347
IRLMG		“IRLMG - IRLM GLOBAL STATUS (IRLM 1.5)” on page 313
IRLMG		“IRLMG - IRLM GLOBAL STATUS (IRLM 2.1 and Later)” on page 353
LATCH	[SORT S0]	“LATCH - Latch Summary” on page 272
LCRES		“LCRES - IRLM Lock Contention by Resource (IRLM 1.5)” on page 321
LCUSR		“LCUSR - IRLM Lock Contention by User (IRLM 1.5)” on page 328
LHRES		“LHRES - IRLM Locks Held by Resources (IRLM 1.5)” on page 330
LHUSR		“LHUSR - IRLM Locks Held by User (IRLM 1.5)” on page 337
LCRES		“LCRES - IRLM Lock Contention by Resource (IRLM 2.1)” on page 362
LCUSR		“LCUSR - IRLM Lock Contention by User (IRLM 2.1)” on page 370
LHRES		“LHRES - IRLM Locks Held by Resources (IRLM 2.1)” on page 372
LHUSR		“LHUSR - IRLM Locks Held by User (IRLM 2.1)” on page 380
LOGST		“LOGST - Log Statistics” on page 279
LUSRD	[regionnum]	“LUSRD - IRLM Lock User Detail (IRLM 1.5)” on page 339
MFSST		“MFSST - MFS Statistics” on page 83

Table 3. Resource Analyzer Service Select Codes (continued)

<b>Service Select Code</b>	<b>Parameter</b>	<b>See</b>
MFSUT	[i teratnnum]	“MFSUT - MFS Pool Utilization” on page 87
PI	[regi onnum]	“PI - Program Isolation” on page 284
POOLC	[pool i d]	“POOLC - Pool Summary (CBT)” on page 288
POOLS	[TYPE=ALL   VAR   FIX]	“POOLS - Pool Summary (Non-CBT Variable and Fixed Pools)” on page 291
PSBUT	[i teratnnum CSA]	“PSBUT, DMBUT - PSB and DMB Pool Utilization” on page 109
QUEST		“QUEST - Queue Statistics” on page 89

Table 3. Resource Analyzer Service Select Codes (continued)

Service Select Code	Parameter	See
REGNS	[, #D<nnn   >nnn   =nnn , AC=nnn   (nnn, . . .) , AGN=xxxxxxxx , CK<nnn   >nnn   =nnn when ' dbctl ugr' delete , C1=nnn , C2=nnn , C3=nnn , C4=nnn , CLA=nnn   (nnn, . . .) , CN<nnn   >nnn   =nnn , D2<nnn   >nnn   =nnn , DD<nnn   >nnn   =nnn , DL<nnn   >nnn   =nnn , DT<nnn   >nnn   =nnn , DY<nnn   >nnn   =nnn , GN<nnn   >nnn   =nnn , GU<nnn   >nnn   =nnn , HN<nnn   >nnn   =nnn , HU<nnn   >nnn   =nnn , IO<nnn   >nnn   =nnn , ID<nnn   >nnn   =nnn , IS<nnn   >nnn   =nnn , I2<nnn   >nnn   =nnn , LO<nnnn   >nnnn   =n nnn , LT=xxxxxxxx , MD<nnn   >nnn   =nnn , MG<nnn   >nnn   =nnn , MI<nnn   >nnn   =nnn , MP<nnn   >nnn   =nnn , NA=xxxxxxxx , NI , OP<nnn   >nnn   =nnn , OT<nnn   >nnn   =nnn , PL<nnnn   >nnnn   =n nnn , PS=xxxxxxxx , QU<nnn   >nnn   =nnn , RE<nnn   >nnn   =nnn , SE<nnn   >nnn   =nnn , S0=cc , SQ<nnnn   >nnnn   =n nnn , ST=xxxxxxxx , T0<nnn   >nnn   =nnn , T2<nnn   >nnn   =nnn , TR=xxxxxxxx , TE<nnn   >nnn   =nnn , TY=xxx , U2<nnn   >nnn   =nnn , US=xxxxxxxx , V=SM DLI DB2]	"REGNS - IMS Regions (No Event Collector Data)" on page 187

Table 3. Resource Analyzer Service Select Codes (continued)

Service Select Code	Parameter	See
REGNS	[,AC=nnn   (nnn, . . ) , AGN=xxxxxxxx , C1=nnn , C2=nnn , C3=nnn , C4=nnn , CLA=nnn   (nnn, . . ) , DL<nnn   >nnn   =nnn , DT<nnn   >nnn   =nnn , EL<nnn   >nnn   =nnn , FP<nnn   >nnn   =nnn , GN<nnn   >nnn   =nnn , GU<nnn   >nnn   =nnn , HN<nnn   >nnn   =nnn , HU<nnn   >nnn   =nnn , ID<nnn   >nnn   =nnn , IS<nnn   >nnn   =nnn , LO<nnnn   >nnnn   =nnn , LT=xxxxxxxx , MD<nnn   >nnn   =nnn , MG<nnn   >nnn   =nnn , MI<nnn   >nnn   =nnn , MP<nnn   >nnn   =nnn , NA=xxxxxxxx , NI , PR<nnnn   >nnnn   =nnn , PS=xxxxxxxx , QU<nnn   >nnn   =nnn , RE<nnn   >nnn   =nnn , S0=cc , ST=xxxxxxxx , TO<nnn   >nnn   =nnn , TR=xxxxxxxx , TY=xxx , US=xxxxxxxx , V= <u>SM</u>  DLI DB2]	“REGNS - IMS Regions (No Event Collector Data)” on page 187
RS	[regionnum]	“RS - Real Storage” on page 303
SCHED	[regionnum]	“SCHED - Scheduling Statistics” on page 112
STAT/ STATR	[regionnum]	“STAT/STATR - System Status” on page 206
VSST		“VSST - VSAM Global Pool Statistics” on page 226
VSST	subpoolnum	“VSST - VSAM Subpool Statistics” on page 229



## Chapter 9. Message Format Service Displays

This chapter describes the displays that show resource status, activity, and performance as transactions are entered from user terminals and are mapped to internal message formats by MFS.

### MFSST - MFS Statistics

BMC SOFTWARE -----				MFS STATISTICS				-----PERFORMANCE MGMT			
SERV ==> MFSST				INPUT	12: 40: 46	INTVL=> 3	LOG=> N	TGT==> IMSxxx			
PARM ==>				SCROLL=> N/A							
MFS POOL SPACE				MFS DATA SET							
49,152	TOTAL	POOL	SPACE	DEVICE	VOLUME	USE	RES	SHR	DEVN		
47,696	DYNAMIC	POOL	SPACE	3390	BAB309	PRI	PRM	SHR	835E		
30 STATIC FRE(S)				26 STATIC FRE(S) ASSIGNED							
0 BLOCK(S) WASHED FOR FRE				1 STATIC FRE(S) ACTIVE							
MFS REQUEST STATISTICS											
391 PRE-FETCH REQUEST(S)				234 PRE-FETCH REQUEST(S) IGNORED							
898 IMMEDIATE REQUEST(S)				0 FREE BLOCK REQUEST(S) IGNORED							
652 FREE BLOCK REQUEST(S)											
25 DIRECTORY READ(S)				0 \$SIMSDIR ENTRY(IES)							
19 IMMEDIATE BLOCK READ(S)				6 REQUEST(S) WITHOUT DIRECTORY ENTRY							
				0 I/O ERROR(S) POINT OR READ							
- - - REQUESTS - - -				- - - - FRE LOCATED ON - - - -							
TYPE		NUMBER		IMMEDIATE		QUEUE		FREE		BLOCK QUEUE	
PRE-FETCH		391		0		0%		207		53%	
IMMEDIATE		898		64		7%		762		85%	
FREE BLOCK		652		652		100%					

**Description:** This display shows the current MFS configuration and MFS activity statistics since IMS restart. These counters are incremented by MFS when measured events occur.

**Select Code:** MFSST

**Parameter:** None

**Field Descriptions:** Each of the fields is shown and described below by display area.

#### Area 1

MFS POOL SPACE  
49,152 TOTAL POOL SPACE  
47,696 DYNAMIC POOL SPACE

This area shows total MFS pool space and the dynamic MFS pool space. The descriptions are arranged in alphabetical order.

#### DYNAMIC POOL SPACE

Pool space available for format blocks.

## TOTAL POOL SPACE

The DYNAMIC POOL SPACE plus the overhead space required by directory blocks, FREs, statistics counters, in-core directory entries, and the like.

### **Area 2**

MFS DATA SET					
DEVICE	VOLUME	USE	RES	SHR	DEVN
3390	BAB309	PRI	PRM	SHR	835E

This area displays information about the first extent of the MFS data set. The descriptions are arranged in alphabetical order.

#### DEVICE

The device type on which the MFS data set is stored. If ?DEVT? appears, it indicates an unknown DASD device type. If ?UNIT? appears, it indicates an unsupported device type. BMC Software should be contacted to include these device types.

#### DEVN

The address of the volume containing the MFS data set.

#### RES

The residency status of the volume containing the MFS data set. RES can be SYS (system resident), PRM (permanently resident), RSV (reserved), or RMV (removable).

#### SHR

Whether the MFS data set is on shared DASD. The column is blank if it is not and SHR if it is.

#### USE

The USE attribute. USE can be either PRI (private), PUB (public), or STO (storage).

#### VOLUME

The name of the volume containing the MFS data set.

### **Area 3**

30 STATIC FRE(S)	26 STATIC FRE(S) ASSIGNED
0 BLOCK(S) WASHED FOR FRE	1 STATIC FRE(S) ACTIVE

This area shows the existing allocation of fetch request elements (FREs). The descriptions are arranged in alphabetical order.

#### BLOCKS WASHED FOR FRE

The number of times since IMS restart that an MFS block was washed from the dynamic pool to release an FRE.

#### STATIC FRE(S)

The total number of FREs, defined at IMS restart.

#### STATIC FRE(S) ACTIVE

The number of assigned FREs being actively used, either prefetch or immediate.

#### STATIC FRE(S) ASSIGNED

The number of FREs actually assigned to controlling a block.

#### **Area 4**

MFS REQUEST STATISTICS	
391 PRE-FETCH REQUEST(S)	234 PRE-FETCH REQUEST(S) IGNORED
898 IMMEDIATE REQUEST(S)	0 FREE BLOCK REQUEST(S) IGNORED
652 FREE BLOCK REQUEST(S)	

This area shows the MFS request statistics accumulated since IMS restart. The descriptions are arranged in alphabetical order.

#### FREE BLOCK REQUEST(S)

The number of times that the MFS pool handler is informed that a format block is no longer actively being referenced and is now a candidate for being washed from the pool if space is needed for another request.

#### FREE BLOCK REQUEST(S) IGNORED

The number of times that a free block request has been ignored because of IMS internal problems.

#### IMMEDIATE REQUEST(S)

The actual need for a block. If the requested block is in the pool, processing continues. If it is not, space must be obtained in the pool and the block must be located and read into the pool from the MFS data set.

#### PRE-FETCH REQUEST(S)

Requests that anticipate future needs for particular format blocks.

#### PRE-FETCH REQUEST(S) IGNORED

Prefetch requests are not required to continue processing and can be ignored by MFS. For example, this counter is incremented when the prefetch request is ignored because no FRE is available. In this case, the prefetch request count is currently not being incremented, causing an apparent imbalance between these two values. This counter is also incremented when PRE-FETCH is turned off.

#### **Area 5**

25 DIRECTORY READ(S)	0 \$\$IMSDIR ENTRY(IES)
19 IMMEDIATE BLOCK READ(S)	6 REQUEST(S) WITHOUT DIRECTORY ENTRY
	0 I/O ERROR(S) POINT OR READ

This area shows the physical I/O required for request processing. The descriptions are arranged in alphabetical order.

#### \$\$IMSDIR ENTRIES

The number of entries in the MFS in-core format index.

#### DIRECTORY READS

The number of directory reads.

#### IMMEDIATE BLOCK READ(S)

The number of immediate block reads.

#### I/O ERROR(S) POINT OR READ

The number of I/O errors.

#### REQUEST(S) WITHOUT DIRECTORY ENTRY

The number of invalid format requests.

<b>Area 6</b>					
- - - REQUESTS - - -			- - - - - FRE LOCATED ON - - - - -		
TYPE	NUMBER		IMMEDIATE	QUEUE	FREE BLOCK QUEUE
PRE- FETCH	391		0	0%	207 53%
I M M E D I A T E	898		64	7%	762 85%
FREE BLOCK	652		652	100%	

This area contains counters, in a matrix format, that are incremented when an FRE is found on any of the internal MFS queues. Each row illustrates a different request type and each column represents a different queue on which the FRE was found.

The descriptions are arranged in alphabetical order.

#### FREE BLOCK

Free block requests.

#### FREE BLOCK QUEUE

The free block queue consists of blocks still in the pool, but not currently being used. The actual count and the percentage of requests located in this queue for each request type are displayed.

#### IMMEDIATE

Immediate requests.

#### IMMEDIATE QUEUE

The immediate queue consists of blocks already in the pool or requests that a block be loaded into the pool immediately. The actual count and the percentage of requests located in this queue for each request type are displayed.

#### NUMBER

The actual count of the request type.

#### PRE-FETCH

Pre-fetch requests.

## MFSUT - MFS Pool Utilization

BMC SOFTWARE ----- MFS POOL UTILIZATION -----					PERFORMANCE MGMT				
SERV ==> MFSUT					INPUT 13: 21: 09 INTVL=> 3 LOG=> N TGT==> IMSxxx				
PARM ==>					SCROLL => N/A				
MFS BLOCK SPACE					MFS FREE SPACE				
BLOCKS	MINIMUM	AVERAGE	MAXIMUM	TOTAL	BLOCKS	MINIMUM	AVERAGE	MAXIMUM	TOTAL
-- CURRENT VALUES --									
97	64	621	5, 512	60, 264	42	8	193	1, 336	8, 128
-- SIMULATED RELEASE --									
95	64	622	5, 512	59, 160	42	8	219	1, 368	9, 232
94	64	622	5, 512	58, 528	42	8	234	2, 000	9, 864
87	64	604	5, 512	52, 592	41	16	385	3, 504	15, 800
82	64	521	3, 565	42, 752	41	16	625	5, 608	25, 640
81	64	520	3, 656	42, 176	40	16	655	7, 520	26, 216
78	64	431	2, 752	33, 648	39	16	890	7, 912	34, 744
64	72	375	2, 248	24, 056	40	16	1, 108	11, 464	44, 336
62	72	354	2, 248	22, 008	41	16	1, 131	12, 888	46, 384
16	104	464	2, 248	7, 432	15	64	4, 064	14, 232	60, 960
13	104	456	2, 248	5, 936	12	64	5, 204	21, 528	62, 456
8	104	599	2, 248	4, 792	8	64	7, 950	24, 808	63, 600
4	104	652	2, 248	2, 608	5	64	13, 156	49, 952	65, 784
2	104	1, 176	2, 248	2, 352	3	64	22, 013	58, 056	66, 040

### Description:

This display shows the current status of the MFS pool: the number of resident blocks, the remaining free space and the fragmentation. The MFS pool is managed by the least-recently-referenced method.

### CURRENT VALUES

This line shows the current status of the pool.

### SIMULATED RELEASE

Each line shows the results of a simulated iteration of the space release algorithm that is used by IMS to free space for a new block. The number of iteration lines displayed depends on the parameter option chosen.

### Select Code:

MFSUT

### Parameter:

Enter 0 to display every iteration of the space release algorithm.

Enter 1 or blank (no entry) to display each iteration that causes an increase in the size of the maximum free space.

Enter 2 through 9 to display every second to ninth increase in maximum free space.

### Field Descriptions:

Following are the field descriptions by "MFS BLOCK SPACE" grouping and "MFS FREE SPACE" grouping. The descriptions are arranged alphabetically within their group:

### **MFS BLOCK SPACE**

AVERAGE	The average block length.
BLOCKS	The number of allocated blocks.
MAXIMUM	The length of the largest allocated block.
MINIMUM	The length of the smallest allocated block.
TOTAL	The total length of all allocated blocks.

### **MFS FREE SPACE**

AVERAGE	The average length of the free spaces in the pool.
BLOCKS	The number of free spaces in the pool.
MAXIMUM	The length of the largest free space in the pool.
MINIMUM	The length of the smallest free space in the pool (indicative of fragmentation).
TOTAL	The total length of all free spaces in the pool.

If less than 18 lines appear, the last line shows the total blocks that are still in use and cannot be released. Therefore, the free space is the maximum available at this point.

## Chapter 10. Queuing Displays

This chapter describes the displays that show resource status, activity, and performance as transactions are queued for processing.

### QUEST - Queue Statistics

BMC SOFTWARE -----												QUEUE STATISTICS -----				PERFORMANCE MGMT							
SERV ==> QUEST				INPUT				13: 21: 09				INTVL=> 3				LOG=> N				TGT==> IMSA			
PARM ==>																				SCROLL => N/A			
QUEUE POOL				NUM				QUEUE DATA SETS															
SIZE		BUFFERS		DD		DS		BLKSIZE		LRECL		DEV		VOLUME		USE		RES		SHR		DEVN	
192, 512		80		QBLKS		1		2, 496		48		3380		BBSYSO		PRI		PRM		SHR		01A0	
				SHORT		10		2, 496		192		3380		BBSYSO		PRI		PRM		SHR		01A0	
				LONG		8		2, 496		2, 496		3380		BBSYSO		PRI		PRM		SHR		01A0	
												QUEUE MANAGER REQUESTS											
101, 123						TOTAL REQUESTS						13, 340						ENQUEUEES					
												2, 940						DEQUEUE/DELETES					
371						REPOSITIONS						1, 379						CANCELS					
												QUEUE BUFFER MANAGER REQUESTS											
52, 878						LOCATES						23, 114						RELEASES					
140, 725						LOCATE & ALTERS						8						PURGES					
												62, 855						TRANSLATIONS					
13, 947						READS						0						WAITS FOR AN AVAILABLE BUFFER					
10, 801						WRITES - TOTAL						115						WAITS FOR OTHER DECB TO READ					
349						WRITES - PURGE						130						WAITS FOR OTHER DECB TO WRITE					
												42						WAITS FOR PURGE					
123						PCBS UNCHAINED						0						WAITS FOR BUFFER ENQ/DEQ					

**Description:** Displays the configuration of the queue data sets, the queue pool, and statistics maintained by the IMS queuing routines since restart.

**Select Code:** QUEST

**Parameter:** None

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>											
				QUEUE DATA SETS							
	NUM										
DD	DS	BLKSIZE	LRECL	DEV	VOLUME	USE	RES	SHR	DEVN		
QBLKS	1	2, 496	48	3380	BBSYSO	PRI	PRM	SHR	01A0		
SHORT	10	2, 496	192	3380	BBSYSO	PRI	PRM	SHR	01A0		
LONG	8	2, 496	2, 496	3380	BBSYSO	PRI	PRM	SHR	01A0		

This area describes the placement of the three queue data sets. The descriptions are arranged in alphabetical order.

**BLKSIZE**

Block size of the queue data sets.

DD

Name of the queue type.

LRECL

Logical record length of the queue data sets.

The following columns in Area 1 represent the information for the first data set of a queue type when more than one data set is defined for that queue type:

DEV

The device type where the queue data set is stored. If ?DEVT? appears, it indicates an unknown DASD device type. If ?UNIT? appears, it indicates an unsupported device type. BMC Software should be contacted to include these device types.

DEVN

The device number of the volume containing the queue data set.

NUM DS

Number of data sets defined for this queue type. This number is 1 for a QBLKS data set and can be up to 10 for short or long message queue data sets.

RES

The residency status of the volume containing the queue data set. RES can be SYS (system resident), PRM (permanently resident), RSV (reserved), or RMV (removable).

SHR

Whether the queue data set is on shared DASD. The column is blank if it is not and SHR if it is.

VOLUME

The name of the volume containing the queue data set.

USE

The USE attribute. USE can be PRI (private), PUB (public), or STO (storage).

**Area 2**

QUEUE POOL
SIZE    BUFFERS
192, 512        80

This area shows queue buffer size and availability. The descriptions are arranged in alphabetical order.

BUFFERS

The number of queue buffers available for use by the IMS queuing routines.

SIZE

The total size of the queue pool.

### Area 3

QUEUE MANAGER REQUESTS	
101, 123 TOTAL REQUESTS	13, 340 ENQUEUES
	2, 940 DEQUEUE/DELETES
371 REPOSITIONS	1, 379 CANCELS

This area shows the number of queue manager requests and the required action. The descriptions are arranged in alphabetical order.

#### CANCELS

Used to complete a series of calls and to inform the queue manager that the message is no longer needed and must be removed without enqueueing.

#### DEQUEUE/DELETES

Informs queue manager that one or more enqueued messages are no longer needed and must be removed from the queues.

#### ENQUEUES

Completes a series of calls and informs the queue manager that the message is to be enqueued on the destination.

#### REPOSITIONS

Requests to relocate a previously examined message segment.

#### TOTAL REQUESTS

The total requests to the queue manager and several particular types of calls.

### Area 4

QUEUE BUFFER MANAGER REQUESTS	
52, 878 LOCATES	23, 114 RELEASES
140, 725 LOCATE & ALTERS	8 PURGES
	62, 855 TRANSLATIONS

This area shows the number of queue manager calls to the queue buffer manager. The descriptions are arranged in alphabetical order.

#### LOCATES

Number of calls to locate a logical record in a queue data set. This is always performed in anticipation of record retrieval.

#### LOCATE & ALTERS

Number of calls to locate and alter a record in a queue data set. This is done when a logical record is to be either allocated or modified.

#### PURGES

Number of purge requests. This causes all altered buffers to be written to their corresponding data set. This is performed for system checkpoint processing.

#### RELEASES

Number of queue data set record releases performed after message dequeue.

#### TRANSLATIONS

Number of main storage address to DRRN (data set location) translations.

#### **Area 5**

13,947 READS  
10,801 WRITES - TOTAL  
349 WRITES - PURGE  
  
123 PCBS UNCHAINED

This area shows the amount of physical I/O activity generated by queuing. Activity to all three queue manager data sets is represented.

The descriptions are arranged in alphabetical order.

#### **PCBS UNCHAINED**

The number of program (communication queue manager) PCBs that lost position because the queue buffer containing the message segment was written out. A queue manager reposition request must be issued before the transaction can continue.

#### **— Tuning Tip —**

If this number is excessive, performance may be enhanced by increasing the queue pool size.

#### **READS**

Total number of reads.

#### **WRITES-PURGE**

Number of writes for purge processing during checkpointing.

#### **— Tuning Tip —**

The difference between number of total writes and the number of writes for purge processing is the number of writes to make space in the pool. If this number is excessive, performance may be enhanced by increasing the queue pool size.

#### **WRITES-TOTAL**

Total number of writes.

#### **Area 6**

0 WAITS FOR AN AVAILABLE BUFFER  
115 WAITS FOR OTHER DECB TO READ  
130 WAITS FOR OTHER DECB TO WRITE  
42 WAITS FOR PURGE  
0 WAITS FOR BUFFER ENQ/DEQ

This area shows an analysis of some of the waits that can occur in queuing. The descriptions are arranged in alphabetical order.

#### **WAITS FOR AN AVAILABLE BUFFER**

Number of waits for an available buffer.

#### **WAITS FOR BUFFER ENQ/DEQ**

Number of waits for buffer enqueues and dequeues.

**WAITS FOR OTHER DECB TO READ**

Number of waits for another DECB to read this buffer.

**WAITS FOR OTHER DECB TO WRITE**

Number of waits for another DECB to write this buffer.

**WAITS FOR PURGE**

Number of waits for purge.



# Chapter 11. Scheduling Displays

This chapter describes the displays that show resource status, activity, and performance as programs are scheduled in the dependent regions to process transactions.

## BALGQ - BALG Queuing

```
BMC SOFTWARE ----- BALGQ QUEUING ----- PERFORMANCE MGMT
SERV ==> BALGQ INPUT 13: 21: 09 INTVL=> 3 LOG=> N TGT==> IMSxxx
PARM ==> <<EXPAND>> SCROLL => N/A
*****QUEUING SUMMARY*****
PROGRAM DBFSAMP3
QUEUED TRANSACTIONS 0
NUMBER OF REGIONS 1

*****REGION SUMMARY*****
REGION 1 2 3 4 5
TYPE MPP M-WFI MPP MPP MDP
STATUS IDLE IDLE IDLE IDLE IDLE
PROGRAM PHDAMINQ DBFSAMP3
CLASS
ENQ TIM
LTERM
PSB SIZ 1, 464 1, 392
QUEUED 0 0
TOT DEQ 264 0
ELAPSED 00: 42: 46 00: 42: 44 00: 05: 55 00: 42: 28 00: 05: 56
```

**Description:** This display presents an overview of the current status of transaction queuing and processing for Fast Path Balancing Groups (BALGs). Up to 12 BALGs can be displayed. If more than 12 are defined, the totals for all additional BALGs are combined with BALG number 12.

**Select Code:** BALGQ

**Parameter:** Region number

The parameter can be specified as:

Single Parameter:  
If only one number is entered, it is used as the starting point of a sequential list of regions. If no parameter is entered, it defaults to the first region.  
  
For example, if regions 1, 2, 3, 5, 6, 7, 8, 9 are active in IMS and you enter:  
  
BALGQ  
6  
  
a region summary of the active regions is displayed in this order:  
  
6 7 8 9 1 2 3 5

Ordered List of Regions:

Regions are listed in the order in which they are to be displayed.

- Individually, separated by commas
- As a range, separated by a dash
- A combination of the two above

For example, if regions 1, 2, 3, 5, 6, 7, 8, 9 are active in IMS and you enter:

BALGQ  
5, 2, 8- 1

a region summary of the active regions is displayed in this order:

5 2 8 9 1

If the starting and ending range are the same, only one region is displayed.

If the ending range is less than the starting range, the list wraps back to region 1 after displaying the highest active region.

Non-Idle Region Specification:

A zero in the list of regions causes only the non-idle regions to be displayed.

For example, if regions 1, 2, 3, 5, 6, 7, 8, 9 are active in IMS and only regions 4 and 5 are processing transactions, and you enter:

BALGQ  
0, 2- 4, 1, 5

a region summary of the active regions is displayed in this order:

4 5

**Expand:**

The DLIST service showing DL/I call details can be invoked for a particular region through cursor selection. Move the cursor to the column that describes the desired region and press ENTER. If the cursor is not in one of the eight columns that pertains to a region, an error message is returned. Use the END PF key to return to the BALGQ display.

DLIST displays the data from the system at the time it is invoked; it is not synchronized with the BALGQ display.

**Field Descriptions:** Each of the fields is shown and described below by display area.

#### Area 1

##### \*\*\*\*\*QUEUEING SUMMARY\*\*\*\*\*

PROGRAM	DBFSAMP3
QUEUED TRANSACTIONS	0
NUMBER OF REGIONS	1

This area shows the status of transaction queuing. The descriptions are arranged in alphabetical order.

#### NUMBER OF REGIONS

The number of message-driven Fast Path regions processing that BALG.

#### PROGRAM

The program name linked to that BALG.

#### QUEUED TRANSACTIONS

The number of transactions queued to that BALG.

#### Area 2

##### \*\*\*\*\*REGION SUMMARY\*\*\*\*\*

REGION	1	2	3	4	5		
TYPE	MPP	M-WFI	MPP	MPP	MDP		
STATUS	IDLE	IDLE	IDLE	IDLE	IDLE		
PROGRAM	PHDAMINQ			DBFSAMP3			
CLASS	1						
ENQ TIME							
LTERM							
PSB SIZ	1,464			1,392			
QUEUED	0			0			
TOT DEQ	264			0			
ELAPSED	00:42:46	00:42:44	00:05:55	00:42:28	00:05:56		

This area shows which transactions are currently being processed in the message regions and whether more are queued. If the status of a region is IDLE, no input belonging to any of its defined classes or its BALG is available.

The descriptions are arranged in alphabetical order.

#### CLASS

The current class for non-Fast Path regions and the transaction routing code for Fast Path regions.

#### ELAPSED

The elapsed time the region has been active (current time - region start time) expressed in hours, minutes and seconds (hh:mm:ss).

#### ENQ TIME

Time when the transaction currently being processed was submitted in hh: mm: ss.

**Note:** This is the time of the originating terminal input transaction for message switches.

## LTERM

LTERM name of the terminal where the transaction being processed was submitted. (This field does not apply to DBCTL threads, region types DBT and ODB.)

## PROGRAM

PSB name of the application program currently being processed.

## PSB SIZE

The total space in bytes used by this program in the PSB pool. This may include the size of the PSB (if not resident), the intent list (if not resident), and a copy of the PDIR (PSB directory entry) if needed for parallel scheduling. If the PSB is resident and not parallel scheduled, this value is zero.

**Note:** This value includes the size of the PSB in the CSA pool.

## QUEUED

Number of currently queued input messages with this transaction code. For an MDP (message-driven program), the number of messages queued on this region's BALG. (This field does not apply to DBCTL threads, region types DBT and ODB.)

## REGION

Region identifier assigned to this region by IMS.

## STATUS

Region status:

ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on).  The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.

OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	Region waiting for an AO message.  Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch).  The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.

WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### TOT DEQ

The total number of messages with this transaction code processed since the last IMS cold start. For MDPs, the number of messages dequeued off this region's BALG. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**Note:** This counter wraps to zero after reaching 32K.

## TYPE

Type of region processing the transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

## CLASQ - Class Queuing

BMC SOFTWARE	-----	CLASS QUEUING	-----	PERFORMANCE MGMT
SERV ==> CLASQ	INPUT	13:21:09	INTVL=> 3	LOG=> N
PARM ==>				TGT==> IMSxxx
				SCROLL => N/A
*****QUEUING SUMMARY*****				
CLASS	001	002	003	004
QUEUED TRANSACTIONS	0	75	2	9
SCHEDULABLE	0	0	2	9
TRANSACTION CODES	0	0	1	3
CLASS	011	012	013	014
QUEUED TRANSACTIONS	0	75	2	9
SCHEDULABLE	0	0	2	9
TRANSACTION CODES	0	0	1	3
*****REGION SUMMARY*****				
REGION	1	2	3	4
TYPE	BMP-WFI	BMP-WFI	MPP	MPP
STATUS	WAITING	ACTV-DLI	IDLE	ACTV-USR
CLS 1/2	.....	..1...2<	..1...2<	..5...6.
CLS 3/4	.....	..3...4.	..3...4.	..7...8.
TRANCD	GIS	WRM	SM421000	DFK00155
ENQ TIM	14:25:44	14:32:40	14:31:48	0:00:00
PRIORITY	0	0	4	6
QUEUED	0	1	2	1
LIMIT			50	1
MSG DEQ	15	1,485	10	0

**Description:** This display presents an overview of the current status of transaction queuing and processing.

**Select Code:** CLASQ

**Parameter:** Region number. The parameter can be specified as:

Single Parameter:

If only one number is entered, it is used as the starting point of a sequential list of regions. If no parameter is entered, it defaults to the first region.

For example, if regions 1, 2, 3, 5, 6, 7, 8, 9 are active in IMS and you enter:

CLASQ  
6

a region summary of the active regions is displayed in this order:

6 7 8 9 1 2 3 5

Ordered List of Regions:

Regions are listed in the order that they are to be displayed.

- Individually, separated by commas
- As a range, separated by a dash
- A combination of the two above

For example, if regions 1, 2, 3, 5, 6, 7, 8, 9 are active in IMS and you enter:

```
CLASQ
5, 2, 8- 1
```

a region summary of the active regions is displayed in this order:

```
5 2 8 9 1
```

If the starting and ending range are the same, only one region is displayed.

If the ending range is less than the starting range, the list wraps back to region 1 after displaying the highest active region.

**Non-Idle Region Specification:**

A zero in the list of regions causes only the non-idle regions to be displayed.

For example, if regions 1, 2, 3, 5, 6, 7, 8, 9 are active in IMS and only regions 4 and 5 are processing transactions, and you enter:

```
CLASQ
0, 2- 4, 1, 5
```

a region summary of the active regions is displayed in this order:

```
4 5
```

**EXPAND:**

The DLIST service showing DL/I call details can be invoked for a particular region through cursor selection. Move the cursor to the column that describes the desired region and press ENTER. If the cursor is not in one of the eight columns that pertains to a region, an error message is returned. Use the END PF key to return to the CLASQ display.

DLIST displays the data from the system at the time it is invoked; it is not synchronized with the CLASQ display.

**Field Descriptions:** Each of the fields is shown and described below by display area.

### Area 1

#### \*\*\*\*\*QUEUING SUMMARY\*\*\*\*\*

CLASS	001	002	003	004	005	006	007	008	009	010
QUEUED TRANSACTIONS	0	75	2	9	1	1	5	23	0	0
SCHEDULABLE	0	0	2	9	1	1	5	15	0	0
TRANSACTION CODES	0	0	1	3	1	1	2	1	0	0
CLASS	011	012	013	014	015	016	017	018	019	020
QUEUED TRANSACTIONS	0	75	2	9	1	1	5	23	0	0
SCHEDULABLE	0	0	2	9	1	1	5	15	0	0
TRANSACTION CODES	0	0	1	3	1	1	2	1	0	0

This area summarizes transaction queuing. The descriptions are arranged in alphabetical order.

#### CLASS

Transaction class.

#### QUEUED TRANSACTIONS

The number of transactions queued.

#### SCHEDULABLE

The number of transactions available for scheduling. (The transaction is not locked, stopped, or priority zero.)

#### TRANSACTION CODES

The number of different transaction codes represented in each class queue (counted only if schedulable).

**Note:** If more than 20 classes are defined, the counts for the excess classes are included with class 20.

### Area 2

#### \*\*\*\*\*REGION SUMMARY\*\*\*\*\*

REGION	1	2	3	4	5	6
TYPE	BMP- WFI	BMP- WFI	MPP	MPP	MPP	BMP
STATUS	WAITING	ACTV- DLI	IDLE	ACTV- USR	SCH- BLR	ACTV- USR
CLS 1/2	.....	.....	.. 1... 2<	.. 1... 2<	.. 5... 6.	.....
CLS 3/4	.....	.....	.. 3... 4.	.. 3... 4.	.. 7... 8.	.....
TRANCD	GIS	WRM	SM421000	DFK00155	LMZ200XX	
ENQ TIM	14: 25: 44	14: 32: 40	14: 31: 48	0: 00: 00		
PRIORITY	0	0		4	6	0
QUEUED	0	1		2	1	0
LIMIT				50	1	
MSG DEQ	15	1, 485		10	0	7

This area shows which transactions are currently being processed in the message regions and what other classes the regions can accept for processing. If the status of a region is IDLE, no input belonging to any of its defined classes is available.

The descriptions are arranged in alphabetical order.

For the following fields, CL1 through CL4, the current active class is shown by a < character following the class:

CL1	The first class this region can process. (This field does not apply for a TYPE of DBT or ODB and for all Fast Path types, including MDP and NDP.)	
CL2	The second class this region can process. (This field does not apply for a TYPE of DBT or ODB and for all Fast Path types, including MDP and NDP.)	
CL3	The third class this region can process. (This field does not apply for a TYPE of DBT or ODB and for all Fast Path types, including MDP and NDP.)	
CL4	The fourth class this region can process. (This field does not apply for a TYPE of DBT or ODB and for all Fast Path types, including MDP and NDP.)	
ENQ TIM	Time when the transaction currently being processed was submitted (hh: mm: ss). <b>Note:</b> This is the time of the originating terminal input transaction for message switches.	
LIMIT	Processing limit count (PROCLIM) of the transaction. Number of transactions that can be processed by this program in one scheduling. If it is unlimited, this field is blank. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
M-DEQ	The number of messages successfully processed by the application program in this scheduling. For MDPs (message-driven program), this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
PRIORITY	Priority of the transaction currently being processed.	
QUEUED	Number of currently queued input messages with this transaction code. For an MDP, this value is the number of messages queued on the region's BALG. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
REGION	Region identifier assigned to this region by IMS.	
STATUS	Region status:  ACTIVE           Active in a nonspecific process.  ACTV-BKO        Region in dynamic backout.  ACTV-DB2        Active in DB2 (Event Collector must be active).  ACTV-DBR        Region active in DBRC.	

ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	<p>Region defined but not started (not yet signed on).</p> <p>The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.</p>
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	<p>Region waiting for an AO message.</p> <p>Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.</p>
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	<p>Region in scheduling (waiting for block mover latch).</p> <p>The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.</p>
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see "LATCH - Latch Summary" on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.

WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### TRANCD

Name of transaction currently being processed.

#### TYPE

Type of region processing the transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

## PSBUT, DMBUT - PSB and DMB Pool Utilization

```

BMC SOFTWARE ----- PSB POOL UTILIZATION ----- PERFORMANCE MGMT
SERV ==> PSBUT  INPUT  13:21:09  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==>
                                SCROLL=> N/A
PSB BLOCK SPACE                PSB FREE SPACE
BLOCKS MINIMUM AVERAGE MAXIMUM TOTAL  BLOCKS MINIMUM AVERAGE MAXIMUM TOTAL
-- CURRENT VALUES --
17      1,336    5,479    9,544 93,152      7      264    1,321    2,536    9,248
-- SIMULATED RELEASE --
16      1,336    5,412    9,544 86,600      7      264    2,257    8,248   15,800
13      1,336    5,097    9,544 66,264      9      336    4,015    9,200   36,136
11      2,984    5,470    9,544 60,176      8     1,416    5,278    9,368   42,224
9       2,984    5,409    9,544 48,688      8     1,416    6,714   20,688   53,712
8       2,984    4,893    6,984 39,144      7     1,416    9,036   27,160   63,256
6       2,984    4,508    6,072 27,048      6     1,416   12,558   35,704   75,352
5       2,984    4,512    6,072 22,560      5     1,552   15,968   41,608   79,840
4       3,120    4,894    6,072 19,576      4     1,552   20,706   51,576   82,824

```

```

BMC SOFTWARE ----- DMB POOL UTILIZATION ----- PERFORMANCE MGMT
SERV ==> DMBUT  INPUT  13:21:09  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==>
                                SCROLL => N/A
DMB BLOCK SPACE                DMB FREE SPACE
BLOCKS MINIMUM AVERAGE MAXIMUM TOTAL  BLOCKS MINIMUM AVERAGE MAXIMUM TOTAL
-- CURRENT VALUES --
29       368      983    2,312 28,512      1    24,736   24,736   24,736   24,736
-- SIMULATED RELEASE --
28       368      997    2,312 27,920      1    25,328   25,328   25,328   25,328
17       576     1,060    2,312 18,032      4      904    8,804   25,696   35,213
15       616     1,122    2,312 16,840      3      904   12,136   34,136   36,408
8        616     1,174    2,312  9,392      2     4,176   21,928   39,680   43,856
7        616     1,077    2,312  7,544      1    45,704   45,704   45,704   45,704
5        616     1,206    2,312  6,032      1    47,216   47,216   47,216   47,216

```

### Description:

These displays show the current status of the PSB and DMB pools, including the number of resident blocks, the free space remaining, and the fragmentation. Because the two pools are managed by the least-recently-referenced method, the display format is the same for both pools (and for the MFS pool).

### Select Code:

PSBUT or DMBUT

### Parameter:

Enter:

- 0        Displays every iteration of the space release algorithm.
- 1        Displays each iteration that causes an increase in the size of maximum free space. Blank, no entry, produces the same result as entering 1.
- 2 to 9   Displays every second to ninth increase in maximum free space.

For PSBUT:

CSA     Displays information about the portion of the PSB pool that resides in CSA if the LSO=S option is used in the control region. This parameter is not valid for other LSO options and will be rejected.

For LSO=S:

blank   The default of no parameter shows the PSB pool information in the DL/I address space. The other LSO options do not have split PSB pools, and PSBUT always shows the entire PSB pool wherever it resides.

**Field Descriptions:** Each of the fields is described below. The descriptions are arranged in alphabetical order.

**CURRENT VALUES**

This line shows the current status of the pool.

**PSB BLOCK SPACE/DMB BLOCK SPACE**

**AVERAGE**

The average block length.

**BLOCKS**

The number of allocated blocks.

**MAXIMUM**

The length of the largest allocated block.

**MINIMUM**

The length of the smallest allocated block.

**TOTAL**

The total length of all allocated blocks.

**PSB FREE SPACE/DMB FREE SPACE**

**AVERAGE**

The average length of the free spaces in the pool.

**BLOCKS**

The number of free spaces in the pool.

**MAXIMUM**

The maximum length of the largest free space in the pool.

**MINIMUM**

The length of the smallest free space in the pool.

**TOTAL**

The total length of all free spaces in the pool.

#### SIMULATED RELEASE

Each line shows the results of a simulated iteration of the space release algorithm that is used by IMS to free space for a new block. The number of iteration lines displayed depends on the parameter option chosen.

If less than 18 lines appear, the last line shows the total blocks that are still in use and cannot be released. Therefore, the free space is the maximum available at this point.

# SCHED - Scheduling Statistics

BMC SOFTWARE ----- SCHEDULING STATISTICS -----				PERFORMANCE MGMT					
SERV ==> SCHED		INPUT 13: 21: 09		INTVL=> 3		LOG=> N		TGT==> IMSxxx	
PARAM ==>								SCROLL => N/A	
SCHEDULING ACTIVITY									
61,417 TOTAL SMBS EXAMINED				19 TOTAL PROGRAM CONFLICTS					
42,877 TOTAL SCHEDULED 70%				45 TOTAL INTENT FAILURES					
18,540 TOTAL FAILURES 30%				18,476 TOTAL OTHER REASONS					
SCHEDULER SEQUENCE QUEUE									
REASON FOR WAIT			RGN	TYPE	CLASSES	CPRTY	PSBNAME	TRANCODE	
WAITING FOR MESSAGE (IDLE)			3	MPP	1				
WAITING FOR INTENT (DB)			4	MPP	3 4	10			
WAITING FOR GU (WFI)			1	BMP			GISPSB	GIS	

- Description:** This display shows the total scheduling activity since IMS restart and the regions currently waiting for scheduling.
- Select Code:** SCHED
- Parameter:** Enter a number in the range 1-nnn (where nnn is a valid region number) to indicate the lowest region number to be displayed on the sequence queue. The default is 1.
- Field Descriptions:** Each of the fields is shown and described below by display area.

Area 1							
SCHEDULING ACTIVITY							
61,417 TOTAL SMBS EXAMINED				19 TOTAL PROGRAM CONFLICTS			
42,877 TOTAL SCHEDULED 70%				45 TOTAL INTENT FAILURES			
18,540 TOTAL FAILURES 30%				18,476 TOTAL OTHER REASONS			

This area shows total scheduling activity. The descriptions are arranged in alphabetical order.

**INTENT FAILURES**

This counter includes actual database intent conflicts and PSB, PSB work, or DMB pool space failures.

## OTHER REASONS

Transaction, program, or database is locked, stopped, or bad (BLDL failed at initialization). Parallel scheduling of a program already processing was permitted, but the transaction failed the load balancing check.

**Note:** Once a stopped transaction is in the message queues, this counter is incremented each time scheduling is attempted for the class; that is, each time another transaction of the same class arrives. This can produce misleading scheduling statistics.

To prevent this counter from being incremented continuously, stopped transactions can be assigned to a processing class that is not associated with an active region.

## PRIORITY CUTOFFS

Scheduling stopped because the defined options allowed only equal and higher priority transactions after an intent failure, and none was available.

## PROGRAM CONFLICTS

Parallel scheduling of a program already processing was not permitted.

## TOTAL FAILURES

The number of unsuccessful attempts at scheduling and the percentage this represents of total activity. The failures are shown by type in the second column of this display area. This is the sum of the halfword counters, with each wrapping to zero if 65,535 is reached.

## TOTAL SCHEDULED

The number of successful attempts at scheduling and the percentage this represents of total activity (TOTAL SMBS EXAMINED - TOTAL FAILURES).

## TOTAL SMBS EXAMINED

This counter is incremented when input is available for a transaction type (enqueued on the SMB) and an attempt is made to schedule the corresponding program.

Area 2							
SCHEDULER SEQUENCE QUEUE							
REASON FOR WAIT	RGN	TYPE	CLASSES	CPRTY	PSBNAME	TRANCODE	
WAITING FOR MESSAGE (IDLE)	3	MPP	1				
WAITING FOR INTENT (DB)	4	MPP	3 4	10			
WAITING FOR GU (WFI)	1	BMP			GISPSB	GIS	

This area lists any regions that are currently waiting for scheduling. The reason for the wait, the region number, the region type, such as MPP (message processing program), BMP (batch message processing), MDP (message-driven program), or FPU (Fast Path utility), and the assigned class are shown. If it is valid for that wait, the program name, transaction code, or cutoff priority could appear.

The descriptions are arranged in alphabetical order. The possible reasons for a wait are:

**WAITING FOR BLOCK MOVER**

The routine that loads the intent lists and the PSB and DMB blocks is only serially reusable.

**WAITING FOR GU (BALG)**

A Fast Path MDP (Message-Driven Program) is enqueued on the BALG waiting for an input message to process.

**WAITING FOR GU (WFI)**

A wait-for-input program is idle.

**WAITING FOR INTENT (DB)**

The region is idle, waiting to update a database exclusively owned by another already scheduled application program.

**WAITING FOR INTENT (DMB POOL)**

Not enough space to load the needed DMBs in the pool.

**WAITING FOR INTENT (PGM)**

Program conflict has occurred because parallel scheduling limit has been reached or the program is not eligible for parallel scheduling.

**WAITING FOR INTENT (PSB POOL)**

Not enough space to load this PSB in the pool or not enough space in the PSB work pool.

**WAITING FOR MESSAGE (IDLE)**

No input of the assigned classes is available.

---

## Chapter 12. Region Displays

These displays show application program activity in the IMS dependent region. You can use them to determine which transactions and application programs are active, how much work they are doing by observing message processing and database access, or how much work they still have to do by watching how many of that type are queuing. You can expand the information by:

- Positioning the cursor on a cursor-sensitive field and pressing ENTER to see more details about that field from a related service
- Using the EXPAND field to select the detailed region display to see what a specific transaction is doing

You can also isolate problem regions by using service parameters that allow you to sort and filter any column. For example, you can use SORT to find the highest number of DL/I calls or the longest elapsed time. Or, you can use a filtering parameter that restricts the display only to those regions that are running specific transactions or a class of transactions.

### — Navigating Regions Data —

When you invoke REGNS, all of the information displayed by the REGNS and DREGN services is collected in a buffer. This allows you to analyze information frozen in time. To refresh the information, press ENTER when the cursor is on the SERV or PARM line.

The regions displays include a:

- REGNS service

There are two versions. If the Event Collector is available, REGNS uses the additional data collected by the Event Collector to provide you with more information about IMS dependent region activity. The Event Collector also allows REGNS to show you IMS dependent region activity for the transactions currently processing rather than just for the duration of a PSB scheduling.

REGNS presents the data by views as follows:

#### **With the Event Collector:**

REGNS provides the following views of IMS dependent region activity for the transactions currently processing:

- Summary View
- Message View
- DL/I View
- DB2 View

**Without the Event Collector:**

REGNS provides the following views of IMS dependent region activity for the duration of a PSB scheduling:

- Summary View
- Message View
- DL/I View

It can be selected from the list of analyzer services (Primary Menu Option 1).

- DREGN service

DREGN shows you what a transaction is doing in a specific region. It can be requested by:

- EXPANDING from the REGNS service to the DREGN service
- Selecting the DREGN service from the list of analyzer services

Like REGNS, the amount of data DREGN can display about a specific region depends upon whether the Event Collector is available and active. If the Event Collector is active, DREGN uses this additional data to provide more information about that region's activity for the transactions currently processing rather than the duration of a PSB scheduling. For example, DB2 activity or the amount of CPU time remaining for that region can be displayed.

DREGN presents:

**With the Event Collector:**

DREGN provides the following information about activity for a specific IMS dependent region for the transactions currently processing:

- DC call activity
- DL/I call activity
- Fast Path activity
- DB2 activity
- PSB/transaction information
- Program isolation activity
- System activity
- Paging activity

**Without the Event Collector:**

DREGN provides the following information about activity for a specific IMS dependent region for the duration of a PSB scheduling:

- DC call activity
- DL/I call activity
- Fast Path activity
- PSB/transaction information
- Program isolation activity
- System activity
- Paging activity

In the REGNS and DREGN displays, each thread is shown as a region, identified by the IMS region (PST) number assigned to it, with a region type of DBT for CICS threads or ODB for ODBA threads. Some of the information shown is related to IMS transaction scheduling and therefore is not valid for DBCTL; these fields are blank or zero.

- STAT/STATR service

STAT/STATR shows the status of the total IMS system, work performed, and resources consumed, indicating possible problem areas. It can be selected from the list of analyzer services.

- DLIST service

DLIST is a DL/I call display that shows information about the active DL/I call processed by a specific region. It can be requested by:

- Cursor selection from the PI, DREGN, CLASQ, BALGQ, or LUSRD service.
- Its service select code from the SERV field of a display with a region number as a parameter.

## DLIST - DL/I Call Status

```
BMC SOFTWARE----- DL/I CALL STATUS ----- PERFORMANCE MGMT
SERV ==> DLIST          INPUT    14: 03: 01  INTVL=> 3  LOG=> N  TGT==>  IMSxxx
PARM ==> 5                LINE      1 OF      16  SCROLL => CSR

RGN:  005          STC:  I15XBMP          PSB:  PTEST02          TRAN:  TTEST02
TYPE:  BMP          STATUS: ACTV#DLI        PGM:  MAXSSA          LTERM:

DLI CALL: GU   (CURRENT)                   DATA CAPTURE EXIT ACTIVE
=====
DB PCB:  CUSTHI DM00  AP    YOHI DAM
        CEECCCCDF44CD4400EFCDD44400000000
        34238944000017000480894140000000004
=====
IO- AREA:      SBCIR11BMPSB          PTEST02          IN01A01          IN01A01 HI
        090000ECCDFFCDDEC444444DECEFF409181506CDFFCFF444444440000CDFFCFF4CC
        10009222399112472200000073523020003F360495011010000000008039501101089
=====
SSA- 1:  CUSTOMER (CUSTID  >          )
        CEEEDDCD4CEECC444600000000000000000000000000000005
        34236459D342394000E00000000000000000000000000000D
=====
SSA- 2:  PRODUCT *- (PRODID  >          )
        DDDCECE4564DDCCC44460000000000005
        79644330COD796494000E0000000000000000000000D
***** END OF DATA *****
```

**Description:**

This service provides details about the parameters on a DL/I call. It can be accessed directly by providing the region number as the parameter, or it can be accessed with cursor selection from the PI, DREGN, CLASQ, BALGQ, or LUSRD service.

The call displayed is for the current DL/I call. If no call is active, DLIST displays the last DL/I call (if any). If the region specified no longer has an active program, only RGN, STC or JOB, TYPE, and STATUS are provided by the display. If the region specified is no longer active, an error message is returned.

**CAUTION**

You should restrict use of this service if your site is concerned about displaying sensitive data. The service can be secured using the ACCESS and PMACC parameters described in *Implementing Security for MAINVIEW Products*.

**Note:** If your IMS is performing well, there are times when details about the DL/I call are not available because the data is too transient. In these cases, an informational message is issued in the scrollable area of the display.

**Select Code:**

DLIST

**Parameter:**

Region number

**Scrolling:**

The service is scrollable so that all data for the call can be viewed without data refresh.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>			
RGN: 005	STC: I15XBMP	PSB: PTEST02	TRAN: TTEST02
TYPE: BMP	STATUS: ACTV#DLI	PGM: MAXSSA	LTERM:
DLI CALL: GU	(CURRENT)	DATA CAPTURE EXIT	ACTIVE

This area shows the scheduling information, status, and the active DL/I call for a region. The descriptions are arranged in alphabetical order.

**DATA CAPTURE EXIT ACTIVE**

This message is displayed if the IMS Data Capture Exit is currently invoked for this region. Otherwise, this field is blank.

**DLI CALL**

DL/I function code (such as GU, GN, or GHU). If the region is not active with a program, this entire line is blank.

**(CURRENT)**

Indicates that the DL/I call is currently being executed.

**(LAST)**

Indicates that this was the last DL/I call made, but it has completed. The status code from DL/I can be found in the PCB area of the display.

**LTERM**

Logical terminal that originated the transaction.

**PGM**

Program name currently executing in region.

**PSB**

PSB name currently active in the region.

**RGN**

Region number.

**STATUS**

Region status:

ACTIVE Active in a nonspecific process.

ACTV-BKO Region in dynamic backout.

ACTV-DB2 Active in DB2 (Event Collector must be active).

ACTV-DBR Region active in DBRC.

ACTV-DLI Region active in DL/I.

ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on).  The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination orabend.
WAIT-AOI	Region waiting for an AO message.  Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch).  The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.

WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.

WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the *IBM System Administration Guide* publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### STC (JOB)

Region started task name (job name).

#### TRAN

Name of the transaction code currently active in the region.

#### TYPE

Region type:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.

MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

Area 2

```

=====
DB PCB:  CUSTHIDM00  AP      YOHI DAM
          CEECCCCDFF44CD4400EFCFCCD44400000000
          34238944000017000480894140000000004
          =====

```

This area shows PCB or AIB-related information if the DL/I call uses a second parameter. The descriptions are arranged in alphabetical order.

The first 70 bytes of the PCB are displayed in hexadecimal dump format with the type of PCB identified as follows:

- DB PCB:            PCB is for a database.
- GSAM PCB:        PCB is for a GSAM database.
- IO PCB:           PCB is the I/O PCB.
- TP PCB:           PCB is for the message queue, but it is not the I/O PCB.

The first 70 bytes of the AIB are displayed if the call uses an AIB. The PCB used to process the request is shown in a section of this area immediately below the AIB.

Area 3

```

=====
IO- AREA:      SBCI R11BMPSB      PTEST02      IN01A01      IN01A01 HI
               090000ECCCDFFCDDEC444444DECEEFF409181506CDFCF444444440000CDFCF4CC
               10009222399112472200000073523020003F360495011010000000008039501101089
               =====

```

If the DL/I call uses a third parameter, this area shows destination or I/O area information. The information shown is determined by the type of call. The descriptions are arranged in alphabetical order.

If the call is a CHNG call, the following appears:

- DEST:            The 8-character destination name specified on the CHNG call (TP or IO PCB only).

For all other types of calls, you see:

- I/O AREA:        First 70 bytes in hexadecimal dump format of the I/O area specified with the call.

[illegible]

If the call is an XRST or CHKP call, the following appears:

First 70 bytes of the nth AREA in hexadecimal dump format as specified on the XRST or CHKP call.

MODNAME:

If the call is a STAT call, this area shows:

The requested 8-byte statistics function for the STAT call.

SSA-n

If the call is a ROLS or SETS call, you will see:

A 4-byte identifier used on the ROLS or SETS call.

## DREGN - Region Detail (Event Collector Data)

```

BMC SOFTWARE-----DETAIL REGION SERVICE-----PERFORMANCE MGMT
SERV ==> DREGN          INPUT    11:44:25 INTVL=> 3 LOG=> N TGT==> IMSA
PARM ==> 1                      LINE 1 OF 56 SCROLL=> CSR
EXPAND: DLIST PI LUSRD CICS DB2 GOTO( DC DLI OTHR FP DB2 PSB PI SYS PAGING)
*EVENT COLLECTOR*                      PF10/11 FOR PREV/NEXT REGION
RGNID.....1             STATUS... ACTV-DLI          ENQ TIME...11:44:24
JOB NAME... MPPRGN2      TRANCODE.. SM4            TRN ELAP... 500ms
TYPE..... MPP           PSB..... SM400600         AGN..... PAYROLL
MSG SWITCH 0            LTERM.... SNLC0050         CLASS... >1 2 3 4
SEQ BF USG 25K          USER..... DN002           DB2STAT... DB2D-CON
                      DB2 AUTHID DN002
-----DC CALL ACTIVITY-----
MSG GU.....8           MSG OTHER...8             PROCLIM... 65535
MSG GN.....0           MSG PURG...0              CMD..... 0
CHKPT.....0           MSG ISRT...7              GET CMD.... 0
-----DL/I CALL ACTIVITY-----
DBNAME      GU   GN   GHU   GHN   GNP   GHNP   REPL   ISRT   DLET   TOTAL   I/O
-----
MYPART1     2    1    0    0    0    0    0    0    0    0    3    3
MYPART2     2    1    0    0    0    0    0    0    0    0    3    2
** TOTAL    4    2    0    0    0    0    0    0    0    0    6    5
-----OTHER CALL ACTIVITY-----
CMD.....1             SETO.....0              APSB..... 0
GCMD.....2           SETS.....3              DPSB..... 0
ICMD.....0           SETU.....0              INIT..... 0
RCMD.....0           ROLB.....0              INQY..... 0
GMSG.....0           ROLS.....0              AUTH..... 0
CHNG.....0           XRST.....0              DB DEQ.... 0
-----FP ACTIVITY-----
NBA.....10            DATABASE AREA          RBA      ENQ
OBA.....10            CUSTDB CUSTDB01      0015BA00 SHR
USED.....2            CUSTDB CUSTDB02      00263300 EXE
                      CUSTADD CUSTADD1     00B19300 SHR
-----DB2 ACTIVITY-----SQL TOT....1
DB2NAME... DB2D        CONTROL...0             INSERTS...0
PLANNAME... PHDAMINQ   DYNAMI C...0             DELETES...0
SEL/FCH... 1           DDL.....0             UPDATES...0
OPENS.....0           OTHER.....0
-----PSB/TRANSACTION-----
PSB SIZE... 148        MODE.....SINGL          CONVERS...
PSB WA.... 2880        SEGMENT...SINGLSEG       SPA LEN...
PROCLIM... 65535       QUEUED.....1           PRIORITY... 4
AVG LENG... 200        CPU REMAIN. 9           TOT DEQ... 3
PRLIM CPU... 10
-----PROGRAM ISOLATION ACTIVITY-----
TEST      Q COMMAND    SHR/UPD    EXCLUSIVE
ENQ        14          0          10          0
DEQ        11          0          7          0
CURQ       3          0          3          0
WAIT       0          0          0          0
-----SYSTEM ACTIVITY-----RGN ELAPSED 01:30:04
EXEC IN... DLI-SAS     TCB CPU... 1,740        DOMAIN.... 1
ASID..... 11          SRB CPU... 234         PERF GR... 13
PRIORITY... 238       CPU SRV... 10968      PERF PD... 1
SWAP..... NON-SWAP    IO SRV... 3555
                      MSO SRV... 6
-----PAGING ACTIVITY-----
WRK IN.... 6          INT PGI... 4,040        VIO PGI... 0
WRK OUT... 25         INT PGO... 2,494        VIO PGO... 0
FRAMES.... 6          INT RCL... 405         VIO RCL... 0
SLOTS..... 43         SWAPCNT... 0           SLT VIO... 4,440
COM PGI... 11,042     COM RCL... 346         STOLEN.... 4,145
***** END OF DATA *****

```

**Description:**

This is the Event Collector version of the detailed regions display. This service shows you what a transaction is doing. For example, you can see which databases this transaction accesses and how often, the number of times a transaction accesses a database and with what type of DL/I call, the number of locks held by a BMP between checkpoints, or a summary of the transaction's resource usage.

You access this detailed region display:

- From the REGNS display
    - Move the cursor to a row (region) in the display and press ENTER.

Detailed information about the region you selected is shown.

  - Move the cursor to DREGN in the EXPAND line of the display and press ENTER.
- Detailed information about the region in the first row of the REGNS display is shown.
- From the Analyzer Service List (Primary Menu Option 1)
    - Select DREGN or REGND and press ENTER.
  - From the SERV field of any display
    - Enter DREGN or REGND with a region ID as the parameter.

**Color:**

If you have a color monitor:

Turquoise	Indicates normal values.
Blue	Indicates an EXPAND or GOTO option that cannot be used.
Yellow	Indicates error messages.
Pink	Highlights the following message when a WFI or pseudo-WFI transaction is waiting for input:
	[ PSEUDO] WFI TRAN WAITING - DATA IS FROM LAST TRANSACTION

**Select Code:**

DREGN or REGND

**Parameter:**

Enter:

nnn

where nnn represents a 1- to 3-character numeric region ID.

You can use the PF10 key to display a previous region or the PF11 key to display the next region.

**Expand:**

The display can EXPAND to other services by moving the cursor to the following fields and pressing ENTER:

DLIST	DL/I Call Status display showing DL/I call details for the displayed region.
PI	Program Isolation display showing pool space and enqueue summaries for the displayed region. If PI is not the IMS lock manager, this field cannot be used. It is blue on a color terminal and low intensity on a monochrome terminal, which indicates that it is not available.
LUSRD	The LUSRD Lock User Detail display showing lock information for the displayed region. If IRLM is not the IMS lock manager, this field is blue on a color terminal and low intensity on a monochrome terminal.

**Hyperlink:**

If MAINVIEW for CICS is installed and the transaction is a CICS transaction, you can link to the TASKXPND service in MAINVIEW for CICS by using the CICS field in the EXPAND line.

If MAINVIEW for DB2 is installed and a transaction has an active DB2 thread, you can link to the DUSER service in MAINVIEW for DB2 by using the DB2 field in the EXPAND line.

**GOTO:**

You also can go to a specific area by moving it to the top of the display. Move the cursor to any of the following GOTO fields in the EXPAND line and press ENTER to move that area to the top of the display:

**Notes:**

- If an area of the display cannot be accessed by this method, the fields are blue on a color terminal and low intensity on a monochrome terminal.
- If there is no activity to be displayed, that area of the display does not appear. For example, if a region does not use Fast Path databases, the FP ACTIVITY area of the display is not shown.

DC	DC CALL ACTIVITY
DLI	DL/I CALL ACTIVITY
FP	FP ACTIVITY
DB2	DB2 ACTIVITY
OTHR	Other types of DL/I calls than those shown by DL/I CALL ACTIVITY
PSB	PSB/TRANSACTION

PI	PROGRAM ISOLATION ACTIVITY
SYS	SYSTEM ACTIVITY
PAGING	PAGING ACTIVITY

**Sorting:** This display cannot be sorted.

**Scrolling:** You can scroll the display by:

- Using PF keys to scroll the display up or down.
- Selecting a GOTO field which selects a display area and scrolls it to the top as described in “GOTO:”.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<u>Area 1</u>									
RGNI D. . . . .	1	STATUS. . . .	ACTV-DLI	ENQ TIME. . .	11: 44: 24				
JOB NAME. . .	MPPRGN2	TRANCODE. .	SM4	TRN ELAP. . .	500ms				
TYPE. . . . .	MPP	PSB. . . . .	SM400600	AGN. . . . .	PAYROLL				
MSG SWITCH	0	LTERM. . . .	SNLC0050	CLASS. . . .	>1 2 3 4 S				
EQ BF USG	25K	USER. . . . .	DN002	DB2STAT. . .	DB2D-			CON	
		DB2 AUTHID	DN002						

This area shows general information about the application program activity in the region for the transaction currently processing. The descriptions are arranged in alphabetical order.

**AGN**  
Name of application group

**CLASS**  
Classes defined for the region. The currently active class is shown by a > character preceding the class. (This field does not apply for a TYPE of DBT or ODB and for all Fast Path types, including MDP and NDP.)

**DB2 AUTHID**  
Authorization ID used to access DB2.

**DB2STAT**  
DB2 connection status for the region. If the region is not connected to DB2, this field is blank. If the region is connected to a DB2 subsystem, the first half of the DB2 status message shows the DB2 subsystem name. The second half of the message shows the connection status:

-CON	IMS connects this region to DB2 if a connection is available and the EXEC parameter SSM (to establish a DB2 connection) is valid.
-SON	The application is signed on and a recovery token is assigned by DB2 if the EXEC parameter SSM is valid, the connection is successful, and the application issues a DB2 request.
-THD	IMS had a thread with DB2 when it processed the EXEC parameter SSM to establish a connection to DB2.

**ENQ TIME**

Time when the transaction currently being processed was submitted (hh:mm:ss). This value is the time of the originating terminal input transaction for message switches. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**JOB NAME**

Job name or started task procedure name of the region.

**LTERM**

LTERM name of the terminal where this transaction was submitted.

**MSG SWITCH**

Number of message switches to generate this transaction. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**PSB**

PSB name of the application program currently being processed.

**RGNID**

Region identifier assigned to this region by IMS.

**SEQ BFR USG**

Sequential buffer usage for this region in kilobytes.

**STATUS**

Region status:

ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.

INACTIVE	<p>Region defined but not started (not yet signed on).</p> <p>The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.</p>
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	<p>Region waiting for an AO message.</p> <p>Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.</p>
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	<p>Region in scheduling (waiting for block mover latch).</p> <p>The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.</p>
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see "LATCH - Latch Summary" on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.

WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### TRANCODE

Name of the transaction currently being processed.

#### TRN ELAP

Current time minus the transaction start time.

**Note:** If you have more than one system clock set, they must be synchronized for TRN ELAP to be accurate.

## TYPE

Type of region processing the transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

## USER

ID of the user who submitted this transaction. For a DBCTL transaction, this value is the ID of the CICS user.

<u>Area 2</u>					
-----DC CALL ACTIVITY-----					
MSG GU. ....	8	MSG OTHER. .	8	PROCLIM. ...	65535
MSG GN. ....	0	MSG PURG. ...	0	CMD. ....	0
CHKPT. ....	0	MSG ISRT. ...	7	GET CMD. ...	0

This area shows the amount of DC call activity incurred by the transaction currently processing. The descriptions are arranged in alphabetical order.

#### CHKPT

Number of checkpoint calls issued by this transaction.

#### CMD

Number of command calls issued by this transaction.

#### GET CMD

Number of GET command calls issued by this transaction.

#### MSG GN

Number of MESSAGE GET NEXT calls issued by this transaction. For MDPs, this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### MSG GU

Number of MESSAGE GET UNIQUE calls issued by this transaction. For MDPs (message-driven program), this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### MSG ISRT

Number of MESSAGE INSERT calls issued by this transaction. For MDPs, this value does not include I/O PCB ISRTs. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### MSG OTHER

Number of DC DL/I calls other than GU, GN, CHKPT, PURG, and ISRT. The number includes system service calls, such as SETS and ROLS, and AO application calls, such as CMD and GCMD. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### MSG PURG

Number of MESSAGE PURGE calls issued by this transaction. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### PROCLIM

Processing limit count (PROCLIM). Number of transactions that can be processed by this program in one scheduling. If there is no processing limit, this field is blank. (This field does not apply to DBCTL threads, region types DBT and ODB.)

<u>Area 3</u>											
-----DL/I CALL ACTIVITY-----											
DBNAME	GU	GN	GHU	GHN	GNP	GHNP	REPL	ISRT	DLET	TOTAL	I/O
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
MYPART1	2	1	0	0	0	0	0	0	0	3	3
MYPART2	2	1	0	0	0	0	0	0	0	3	2
** TOTAL	4	2	0	0	0	0	0	0	0	6	5

This area shows the amount of DL/I call activity incurred by the transaction currently processing. The descriptions are arranged in alphabetical order.

#### \*\*TOTAL

Total number of DL/I database calls and I/Os issued by this transaction. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### DBNAME

DL/I database name.

#### DLET

Number of DELETE calls issued by this transaction against the database.

#### GHN

Number of GET HOLD NEXT calls issued by this transaction against the database.

#### GHNP

Number of GET HOLD NEXT within PARENT calls issued by this transaction against the database.

#### GHU

Number of GET HOLD UNIQUE calls issued by this transaction against the database.

#### GN

Number of GET NEXT calls issued by this transaction against the database.

#### GNP

Number of GET NEXT within PARENT calls issued by this transaction against the database.

#### GU

Number of GET UNIQUE calls issued by this transaction against the database.

#### I/O

Amount of I/O issued by this transaction against the database.

#### ISRT

Number of INSERT calls issued by this transaction against the database.

#### REPL

Number of REPLACE calls issued by this transaction against the database.

## TOTAL

Total number of DL/I database calls issued by this transaction against the database:

GU + GN + GHU + GHN + GHNP + REPL + ISRT + DLET + OTHER

This value includes calls to MSDBs and DEDBs and all other calls that are not database calls. (This field does not apply to DBCTL threads, region types DBT and ODB.)

<u>Area 4</u>					
-----OTHER CALL ACTIVITY-----					
CMD. ....	1	SET0. ....	0	APSB. ....	0
GCMD. ....	2	SETS. ....	3	DPSB. ....	0
ICMD. ....	0	SETU. ....	0	INIT. ....	0
RCMD. ....	0	ROLB. ....	0	INQY. ....	0
GMSG. ....	0	ROLS. ....	0	AUTH. ....	0
CHNG. ....	0	XRST. ....	0	DB DEQ. ...	0

This area shows the amount of IMS call activity, other than the calls shown by the “DL/I CALL ACTIVITY” section, incurred by the transaction currently processing. The descriptions are arranged in alphabetical order.

## APSB

Number of ALLOCATE PSB calls issued by this transaction.

## AUTH

Number of AUTHORIZATION calls issued by this transaction.

## CHNG

(This field does not apply for a TYPE of DBT or an IMS DBCTL system)

Number of CHANGE DC calls issued by this transaction.

## CMD

Number of AO COMMAND calls issued by this transaction.

## DB DEQ

Number of DATABASE DEQUEUE calls issued by this transaction.

## DPSB

Number of DEALLOCATE PSB calls issued by this transaction.

## GCMD

Number of AO application GET COMMAND calls issued by this transaction.

## GMSG

Number of AO application GET MESSAGE calls issued by this transaction.

## ICMD

Number of AO application ICMD calls issued by this transaction.

## INIT

Number of INIT calls issued by this transaction.

INQY  
Number of INQUIRY calls issued by this transaction.

RCMD  
Number of RETRIEVE COMMAND calls issued by this transaction.

ROLB  
Number of ROLLBACK calls issued by this transaction.

ROLS  
Number of ROLLBACK TO SETS/SETU calls issued by this transaction.

SETO  
Number of SET OPTIONS calls issued by this transaction.

SETS  
Number of SET BACKOUT POINT calls issued by this transaction.

SETU  
Number of SET UNCONDITIONAL calls issued by this transaction.

XRST  
Number of EXTENDED RESTART calls issued by this transaction.

<u>Area 5</u>					
----- FP ACTIVITY -----					
NBA. ....	10	DATABASE	AREA	RBA	ENQ
OBA. ....	10	CUSTDB	CUSTDB01	0015BA00	SHR
USED. ....	2	CUSTDB	CUSTDB02	00263300	EXE
		CUSTADD	CUSTADD1	00B19300	SHR

This area shows the region Fast Path call activity for the transaction currently processing. The descriptions are arranged in alphabetical order.

NBA  
Number of buffers defined for normal buffer allocation.

OBA  
Number of buffers defined for overflow buffer allocation.

USED  
Total buffers used. If this total is greater than the NBA value, the extra buffers are taken from the overflow buffer allocation.

The following fields display the first 10 Fast Path database locks held by this region. They show the Fast Path database activity occurring for this transaction.

AREA  
Fast Path database area being accessed by this transaction.

DATABASE  
Fast Path database being accessed by this transaction.

ENQ

ENQ type, which can be either SHR (read) or EXE (update).

RBA

IMS resource ID being locked. For Fast Path DEDBs, this value is the high-order three bytes of the control interval RBA.

Area 6					
DB2 ACTIVITY			SQL TOT. . . .		
DB2NAME. . . DB2D	CONTROL. . .	0	INSERTS. . .		0
PLANNAME. . . PHDAMI NQ	DYNAMI C. . .	0	DELETES. . .		0
SEL/FCH. . . .	DDL. . . . .	0	UPDATES. . .		0
OPENS. . . . .	OTHER. . . . .	0			

This area shows the region DB2 subsystem call activity for the transaction currently processing. The descriptions are arranged in alphabetical order.

For DBCTL, only BMPs or JBP are shown. CICS does not use IMS services to access DB2.

CONTROL

Total number of SQL control type calls issued to DB2 (GRANT or REVOKE, for example) by the transaction currently processing.

DB2NAME

DB2 subsystem ID.

DDL

Number of SQL Data Definition Language calls issued to DB2 (CREATE, DROP, ALTER, COMMENT, or LABEL, for example) by the transaction currently processing.

DELETES

Total number of SQL DELETE calls issued to DB2 by the transaction currently processing.

DYNAMIC

Total number of SQL dynamic calls issued to DB2 (PREPARE, DESCRIBE, or EXECUTE, for example) by the transaction currently processing.

INSERTS

Total number of SQL INSERT calls issued to DB2 by the transaction currently processing.

OPENS

Total number of SQL OPEN cursor calls issued to DB2 by the transaction currently processing.

OTHER

Number of other SQL calls issued to DB2 by the transaction currently processing that do not fit in any of the other types defined in this display (EXPLAIN, LOCK, LABEL, CLOSE, or table and security manipulation, for example).

PLANNAME

Plan name used for this transaction.

#### SEL/FCH

Total number of SQL SELECT and FETCH calls issued to DB2 by the transaction currently processing.

#### SQL TOT

Total number of SQL calls issued to DB2 by the transaction currently processing.

#### UPDATES

Total number of SQL UPDATE calls issued to DB2 by the transaction currently processing.

<u>Area 7</u>			
-----PSB/TRANSACTION-----			
PSB SIZE. . .	148	MODE. . . . .	SNGL
PSB WA. . . .	2880	SEGMENT. . . .	SNGLSEG
PROCLIM. . .	65535	QUEUED. . . .	1
AVG LENG. . .	200		
PRLIM CPU. .	10	CPU REMAIN.	9
		CONVERS. . . .	
		SPA LEN. . . .	
		PRIORITY. . . .	4
		TOT DEQ. . . .	3

This area shows general information about the application program activity and the transaction currently processing. The descriptions are arranged in alphabetical order.

#### AVG LENG

The average length of all input messages received with this transaction code (as calculated by the IMS queue manager). For MDPs, this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### CONVERS

Indicates whether this transaction is a conversational transaction (YES) or not (NO). It is blank for Fast Path regions.

#### CPU REMAIN

Remaining CPU time in seconds for this transaction before the CPU processing limit (PRLIM CPU) is reached.

#### MODE

Transaction mode, which can be SNGL or MULT. An IMS synchronization point occurs at each request for a new message (SNGL) or only at program termination (MULT). (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### PRIORITY

Current scheduling priority of this transaction.

#### PRLIM CPU

Processing limit CPU time in seconds.

#### PROCLIM

Processing limit count. The number of transactions that can be processed by this program in one scheduling. If there is no limit, the field is blank.

#### PSB SIZE

Total space in bytes used by this program in the PSB pool. This may include the size of the PSB (if not resident), the intent list (if not resident), and a copy of the PDIR (PSB directory entry) if needed for parallel scheduling. If the PSB is resident and not parallel-scheduled, this value is zero.

**Note:** This value includes the size of the PSB in the CSA pool.

PSB WA

Size of the work area required by this program in the PSB work pool.

QUEUED

Number of currently queued input messages for this transaction code. For MDPs (message-driven program), it is the number of messages queued on this region's BALG. (This field does not apply to DBCTL threads, region types DBT and ODB.)

SEGMENT

SNGLSEG for single segment or MULTSEG for multiple segment. (This field does not apply to DBCTL threads, region types DBT and ODB.)

SPA LEN

Length of a conversational transaction's SPA (scratch pad area). If the transaction code is not conversational, this field is blank.

TOT DEQ

Total number of messages processed for this transaction code since IMS cold start. For MDPs, the number of messages dequeued from this region's BALG.

**Note:** This counter wraps to zero after reaching 32K.

Area 8				
----- PROGRAM ISOLATION ACTIVITY -----				
	TEST	Q COMMAND	SHR/UPD	EXCLUSIVE
ENQ	14	0	10	0
DEQ	11	0	7	0
CURQ	3	0	3	0
WAIT	0	0	0	0

This area shows IMS program isolation (PI) statistics for the transaction currently running.

These statistics give you the amount of segment-locking activity an application is doing. They are not reset until the transaction terminates. If the application processes multiple transactions in one scheduling, these statistics include the activity of transactions processed earlier in the scheduling.

The columns in this area of the display show the IMS request type. The rows show the type of activity. To interpret the data in this area of the display, look at the row to see the type of activity, then look at the column for the request type. For example, Row 1, Column 1 shows the number of TEST ENQs and Row 2, Column 3 shows the number of SHR or UPD DEQs.

The request types are:

TEST

Test enqueues are issued by IMS to determine if a resource is currently held by another task. Test enqueues do not result in a lock being acquired. They may optionally request to wait for a resource to become available before control is returned to them.

#### Q Command

These are segment reservation requests made by the application program using the Q command code in the segment search argument (SSA).

#### SHR/UPD

These are normal enqueue requests to lock a database segment. SHR locks are for reads (with integrity) and UPD locks are for updates, deletes, and inserts. The statistics for SHR and UPD are not reported separately, because lock requests are combined into one number by IMS.

#### Exclusive

These are exclusive enqueues. They are issued by IMS administrative tasks that are not related to database segment level requests. Examples of these types of requests are data set opens and extends. They also are used for Fast Path databases whenever two or more regions need a particular control interval or the use of the overflow buffers.

The types of activity reported include:

#### ENQ

Number of enqueue requests since transaction code scheduling to acquire a resource lock.

#### DEQ

Number of dequeue requests since transaction code scheduling to release a resource lock.

#### CURQ

Number of current enqueue requests calculated by subtracting the number of dequeues from the number of enqueues. This only approximates the number of currently held enqueues, because:

- Some dequeue requests only demote a lock from UPD (update) to SHR (share). The request is counted as a dequeue, but the application still holds a lock.
- At synchronization point time, all enqueues are released and are not counted as dequeues. These statistics are not reset until the application program terminates. Thus, the CURQ field may be too high if multiple transactions are processed in the same scheduling.
- The exclusive dequeue count can be larger than the exclusive enqueue count, because the dequeue count at synchronization point may include resources that were not counted as enqueues.

## WAIT

Number of times the application had to wait since transaction code scheduling for an enqueue to be granted.

### Tuning Tip

To identify an application doing a database scan, look for increasing counts in the ENQ and DEQ rows and no change in count in the CURQ row for the SHR/UPD column.

### Area 9

```
-----SYSTEM ACTIVITY-----RGN ELAPSED 01:30:04
EXEC IN... DLI-SAS          TCB CPU... 1,740      DOMAIN.... 1
ASID..... 11              SRB CPU... 234        PERF GR... 13
PRI ORITY.. 238           CPU SRV... 10968      PERF PD... 1
SWAP..... NON-SWAP        IO SRV... 3555
                               MSO SRV... 6
```

This area shows general information about the region in relation to the operating system. The descriptions are arranged in alphabetical order.

## ASID

OS/390 address space identification number.

## CPU SRV

Total CPU service units for this address space.

## DOMAIN

Address space domain used by SRM. N/A is shown if WLM is in goal mode.

## EXEC IN

Execution is in the control or dependent region:

ALC	Currently executing in a TCB for IMS internal use
ALM	Currently executing in a selective dispatching TCB (application program maintaining LU 6.2 conversation for a long time)
BMP	Currently executing under a batch message processing region TCB
CTL	Currently executing in the control region TCB
CTX	Currently executing under a control region auxiliary TCB
DBT	Currently executing under a DBCTL (CICS attachment) region TCB
DCC	Currently executing under a DCCTL management TCB
DLG	Currently executing in the DASD log TCB
DLI	Currently executing in a TCB in the DL/I address space
DRA	Currently executing under a database resource adapter thread (DBCTL connection) TCB
DRC	Currently executing in a database recovery processing TCB
DTT	Currently executing in a TCB for IMS internal use

DYA	Currently executing in a dynamic allocation TCB
DYD	Currently executing in a TCB for IMS internal use
DYS	Currently executing in a dynamic control services TCB (dynamic allocation service in the DL/I subordinate address space)
ESI	Currently executing in an external subsystem interface TCB
ESS	Currently executing an external subsystem request TCB
IFP	Currently executing under a Fast Path region TCB
JBP	Currently executing under a Java batch message processing TCB
JMP	Currently executing under a Java message processing TCB
LRD	Currently executing in a TCB for IMS internal use
LRM	Currently executing in a TCB for IMS internal use
LSD	Currently executing in a local storage management TCB
LUM	Currently executing in an APPC request management TCB
MOD	Currently executing in an IMS control, MODIFY TCB
MPP	Currently executing under a message processing region TCB
ODM	Currently executing under the ODBA syncpoint mother TCB
ODS	Currently executing under an ODBA syncpoint daughter TCB
OII	Currently executing under the OTMA initialization TCB
OIC	Currently executing under the OTMA control TCB
OIM	Currently executing under an OTMA member TCB
RCF	Currently executing under the allocate RCF TCB
RDS	Currently executing in a restart data set processing TCB
RLM	Currently executing in an IRLM request processing TCB
RST	Currently executing in a restart processing TCB
RWD	Currently executing in the online recovery service daughter TCB
RWM	Currently executing in the online recovery service main TCB
SQ1	Currently executing in the shared queues SQ1 TCB
SQ2	Currently executing in the shared queues SQ2 TCB
STC	Currently executing in a storage compression TCB
STM	Currently executing in a storage management TCB (job step TCB)
TMC	Currently executing in a time controlled option (TCO) TCB
TRA	Currently executing in an IMS trace processing TCB
XCF	Currently executing in an Extended Communication Facility (LU 6.2) TCB
XFP	Currently executing in a Fast Path common services TCB

xx=?            Displayed when an unknown task ID (TCB) is encountered

                 The unknown task ID replaces xx (contact BMC Software Customer Support).

IO SRV            Total I/O service units for this address space.

MSO SRV            Total main storage service units for this address space.

PERF GR            Address space performance group used by SRM. N/A is shown if WLM is in goal mode.

PERF PD            Address space current performance period used by SRM. N/A is shown if WLM is in goal mode.

PRIORITY            Current priority of this address space on the dispatch queue. N/A is shown if WLM is in goal mode.

RGN ELAPSED            Elapsed time the region has been active (current time - region start time) expressed in hours, minutes, and seconds (hh:mm:ss).

SRB CPU            Total region SRB CPU time expressed in seconds.

SWAP            Current swapping status of the address space:

IN	Swapped in
NON-SWAP	Nonswappable
OUT	Swapped out

TCB CPU            Total region TCB CPU time expressed in seconds.

<u>Area 10</u>					
----- PAGING ACTIVITY-----					
WRK OUT. . . .	25	INT PGO. . . .	2, 494	VIO PGO. . . .	0
FRAMES. . . .	6	INT RCL. . . .	405	VIO RCL. . . .	0
SLOTS. . . . .	43	SWAPCNT. . . .	0	SLT VIO. . . .	4, 440
COM PGI. . . .	11, 042	COM RCL. . . .	346	STOLEN. . . .	4, 145
***** END OF DATA *****					

This area shows a summary of paging activity in the region. The descriptions are arranged in alphabetical order.

Paging counts refer to the current SRM interval (time since last swap-in). These count fields are blank when the corresponding address space is swapped out. When the region is nonswappable, the counts are totals accumulated since the region started.

COM PGI	Number of page-ins in the common area (CSA/LPA) for this address space during this interval.
COM RCL	Number of page reclaims in the common area (CSA/LPA) for this address space during this interval.
FRAMES	Number of page frames of real storage currently allocated to this address space.
INT PGI	Number of page-ins during this interval.
INT PGO	Number of page-outs during this interval.
INT RCL	Number of page reclaims during this interval.
SLOTS	Number of pages (slots) of direct access storage on the paging device that is allocated to this address space. The number includes VIO and non-VIO slots.
SLT VIO	Number of pages (slots) of direct access storage on the paging devices that are allocated to this address space for VIO.
STOLEN	Number of pages stolen for this address space during this interval.
SWAPCNT	Number of times this address space has been swapped out. Even though the regions are marked nonswappable, they may swap during initialization. After this, the count remains the same.
VIO PGI	Number of page-ins for VIO during this interval.
VIO PGO	Number of page-outs for VIO during this interval.
VIO RCL	Number of VIO page reclaims during this interval.
WRK IN	Number of page frames last swapped in.
WRK OUT	Number of page frames last swapped out.

## DREGN - Region Detail (No Event Collector Data)

```

BMC SOFTWARE----- IMS DETAIL REGION ----- PERFORMANCE MGMT
SERV ==> DREGN INPUT 10:42:38 INTVL=> 3 LOG=> N TGT==> IMSA
PARAM ==> 002 ROW 1 OF 40 SCROLL=> CSR
EXPAND: DLI ST PI LUSRD CI CS DB2 GOTO( DC DLI OTHR FP DB2 PSB PI SYS PAGI NG)
PF10/11 FOR PREV/NEXT REGION
RGNID..... 2 STATUS.... ACTV-USR ENQ TIME.. 10:42:36
JOB NAME.. MPPRGN2 TRANCODE.. DLZZ TRN ELAP.. 200ms
TYPE..... MPP PSB..... STBCUSR AGN..... PAYROLL
MSG SWITCH 0 LTERM.... T002 CLASS. >1 3 5 2
SEQ BF USG 0 USER..... CIR11 DB2STAT... DB2D- CON
-----DC CALL ACTIVITY-----
MSG GU... 0 MSG OTHER. 0 PROCLIM... 0
MSG GN... 0 MSG PURG.. 0 CMD..... 0
CHKPT.... 0 MSG ISRT.. 0 GET CMD... 0
-----DL/I CALL ACTIVITY----- DL/I TOT.. 10
DB GU.... 1 DB GNP.... 0 TRAN PROC. 0
DB GHU.... 0 DB GHNP... 0 DB ISRT... 0
DB GN.... 9 DEDB..... 0 DB REPL... 0
DB GHN.... 0 MSDB..... 0 DB DLET... 0
-----OTHER CALL ACTIVITY-----
CMD..... 1 SETO..... 0 APSB..... 0
GCMD..... 2 SETS..... 3 DPSB..... 0
ICMD..... 0 SETU..... 0 INIT..... 0
RCMD..... 0 ROLB..... 0 INQY..... 0
GMSG..... 0 ROLS..... 0 AUTH..... 0
CHNG..... 0 XRST..... 0 DB DEQ... 0
-----FP ACTIVITY-----
NBA..... 10 DATABASE AREA RBA ENQ
OBA..... 10 CUSTDB CUSTDB01 0015BA00 SHR
USED..... 2 CUSTDB CUSTDB02 00263300 EXE
CUSTADD CUSTADD1 00B19300 SHR
-----PSB/TRANSACTION-----
PSB SIZE.. 968 MODE..... SNGL CONVERS... NO
PSB WA... 2520 SEGMENT... SNGLSEG SPA LEN... 0
PROCLIM... 0 QUEUED... 0 PRIORITY... 0
AVG LENG... 200 TOT DEQ... 0
-----PROGRAM ISOLATION ACTIVITY-----
TEST Q COMMAND SHR/UPD EXCLUSIVE
ENQ 0 0 10 1
DEQ 0 0 9 1
CURQ 0 0 1 0
WAIT 0 0 0 0
-----SYSTEM ACTIVITY----- RGN ELAPSED 01:30:04
EXEC IN... DLI-SAS TCB CPU... 1,740 DOMAIN.... 1
ASID..... 11 SRB CPU... 234 PERF GR... 13
PRIORITY.. 238 CPU SRV... 10968 PERF PD... 1
SWAP..... NON-SWAP IO SRV... 3555
MSO SRV... 6
-----PAGING ACTIVITY-----
WRK IN... 0 INT PGI... 0 VIO PGI... 0
WRK OUT... 0 INT PGO... 0 VIO PGO... 0
FRAMES... 0 INT RCL... 0 VIO RCL... 0
SLOTS.... 0 SWAPCNT... 1 SLT VIO... 0
COM PGI... 5 COM RCL... 0 STOLEN... 5434
***** END OF DATA *****

```

### Description:

This version of DREGN shows you what transaction is currently running and the cumulative DL/I statistics and total resource usage since the PSB was scheduled in a dependent region. You can use the statistics to see region activity details about a transaction such as transaction arrival time into the input queue and elapsed time, total

number of DL/I or Fast Path calls issued since PSB scheduling, and a summary of paging and operating system activity for that region.

The DREGN version shown in “DREGN - Region Detail (Event Collector Data)” on page 125 uses the information produced by the Event Collector to provide region activity details about the work being performed by the transaction currently running. Those details comprise resource usage statistics for that transaction instance rather than cumulative since PSB scheduling. The statistics from that version are more extensive, including DB2 data, and the version is available only when the Event Collector is active.

You access this detailed region display:

- From the REGNS display
  - Move the cursor to a row (region) in the display and press ENTER.

Detailed information about the region you selected is shown.

- Move the cursor to DREGN in the EXPAND line of the display and press ENTER.

Detailed information about the region in the first row of the REGNS display is shown.

- From the Analyzer Service List (Primary Menu Option 1)
  - Select DREGN or REGND and press ENTER.
- From the SERV field of any display.
  - Enter DREGN or REGND with a region ID as the parameter.

**Color:**

If you have a color monitor:

Turquoise	Indicates normal values.
Blue	Indicates an EXPAND or GOTO option that cannot be used.
Yellow	Indicates error messages.
Pink	Highlights the following message when a WFI or pseudo-WFI transaction is waiting for input:  [PSEUDO] WFI TRAN WAITING - DATA IS FROM LAST TRANSACTION

**Select Code:**

DREGN or REGND

**Parameter:**

Enter:

nnn

where nnn represents a 1- to 3-character numeric region ID.

You can use the PF10 key to display a previous region or the PF11 key to display the next region.

**Expand:**

The display can EXPAND to other services by moving the cursor to the following fields and pressing ENTER:

DLIST	DL/I Call Status display showing DL/I call details for the displayed region.
PI	Program Isolation display showing pool space and enqueue summaries for the displayed region. If PI is not the IMS lock manager, this field cannot be used. It is blue on a color terminal and low intensity on a monochrome terminal, which indicates that it is not available.
LUSRD	The LUSRD Lock User Detail display showing lock information for the displayed region. If IRLM is not the IMS lock manager, this field is blue on a color terminal and low intensity on a monochrome terminal.

**Hyperlink:**

If MAINVIEW for CICS is installed and the transaction is a CICS transaction, you can link to the TASKXPND service in MAINVIEW for CICS by using the CICS field in the EXPAND line.

If MAINVIEW for DB2 is installed and a transaction has an active DB2 thread, you can link to the DUSER service in MAINVIEW for DB2 by using the DB2 field in the EXPAND line.

**GOTO:**

You also can go to a specific area by moving it to the top of the display. Move the cursor to any of the following GOTO fields in the EXPAND line and press ENTER:

**Notes:**

- If an area of the display cannot be accessed by this method, the fields are blue on a color terminal and low intensity on a monochrome terminal.
- If there is no activity to be displayed, that area of the display does not appear. For example, if a region does not use Fast Path databases, the FP ACTIVITY area of the display is not shown.

DC	DC CALL ACTIVITY
----	------------------

DLI	DL/I CALL ACTIVITY
-----	--------------------

FP	FP ACTIVITY
OTHR	Other types of DL/I calls than those shown by DL/I CALL ACTIVITY
PSB	PSB/TRANSACTION
PI	PROGRAM ISOLATION ACTIVITY
SYS	SYSTEM ACTIVITY
PAGING	PAGING ACTIVITY

**Sorting:** This display cannot be sorted.

**Scrolling:** You can scroll the display by:

- Using PF keys to scroll the display up or down.
- Selecting a GOTO field which selects a display area and scrolls it to the top as described in “GOTO:”.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<u>Area 1</u>			
RGNI D. . . . .	2	STATUS. . . . .	ACTV-USR
JOB NAME. . .	MPPRGN2	TRANCOD. . .	DLZZ
ENQ TIME. . .			10: 42: 36
		TRN ELAP. . .	200ms
TYPE. . . . .	MPP	PSB. . . . .	STBCUSR
MSG SWITCH	0	AGN. . . . .	PAYROLL
SEQ BF USG	0	LTERM. . . .	T002
		CLASS. . . .	>1 3 5 2
		USER. . . . .	C1R11
		DB2STAT. . .	DB2D- CON

This area shows general information about the application program activity in the region for the transaction currently processing. The descriptions are arranged in alphabetical order.

**AGN**  
Name of the application group assigned to this region (if any).

**CLASS**  
Classes defined for the region. The currently active class is shown by a > character preceding the class. (This field does not apply for a TYPE of DBT or ODB and for all Fast Path types, including MDP and NDP.)

**DB2STAT**  
DB2 connection status for the region. If the region is not connected to DB2, this field is blank. If the region is connected to a DB2 subsystem, the first half of the DB2 status message shows the DB2 subsystem name. The second half of the message shows the connection status:

-CON      IMS connects this region to DB2 if a connection is available and the EXEC parameter SSM (to establish a DB2 connection) is valid.

- SON      The application is signed on and a recovery token is assigned by DB2 if the EXEC parameter SSM is valid, the connection is successful, and the application issues a DB2 request.
- THD      IMS had a thread with DB2 when it processed the EXEC parameter SSM to establish a connection to DB2.

#### ENQ TIME

Time when the transaction currently being processed was submitted (hh:mm:ss). this value is the time of the originating terminal input transaction for message switches. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### JOB NAME

Job name or started task procedure name of the region.

#### LTERM

LTERM name of the terminal where this transaction was submitted.

#### MSG SWITCH

Number of message switches to generate this transaction. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### PSB

PSB name of the application program currently being processed.

#### RGNID

Region identifier assigned to this region by IMS.

#### SEQ BF USG

Sequential buffer usage for this region in kilobytes.

#### STATUS

Region status:

- ACTIVE      Active in a nonspecific process.
- ACTV-BKO      Region in dynamic backout.
- ACTV-DB2      Active in DB2 (Event Collector must be active).
- ACTV-DBR      Region active in DBRC.
- ACTV-DLI      Region active in DL/I.
- ACTV-MQS      Region active in MQSeries (Event Collector must be active).
- ACTV-SCH      Region active in nonspecific CREATE THREAD process.
- ACTV-USR      Region active in the application program.
- IDLE      Region waiting for non-WFI input to process.
- IDLE-HOT      Pseudo-WFI region waiting for input from the same transaction.

IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	<p>Region defined but not started (not yet signed on).</p> <p>The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.</p>
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	<p>Region waiting for an AO message.</p> <p>Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.</p>
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	<p>Region in scheduling (waiting for block mover latch).</p> <p>The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.</p>
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see "LATCH - Latch Summary" on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).

WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more

information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### TRANCODE

Name of the transaction currently being processed.

#### TRN ELAP

Current time minus the transaction start time.

**Note:** If you have more than one system clock set, they must be synchronized for TRN ELAP to be accurate.

#### TYPE

Type of region processing the transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.

NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

USER  
 ID of the user who submitted this transaction. For a DBCTL transaction, this value is the ID of the CICS user.

<u>Area 2</u>					
-----DC CALL ACTIVITY-----					
MSG GU. . . .	0	MSG OTHER.	0	PROCLIM . .	0
MSG GN. . . .	0	MSG PURG. .	0	CMD. . . . .	0
CHKPT. . . .	0	MSG ISRT. .	0	GET CMD. . .	0

This area shows the amount of DC call activity incurred by the transaction currently processing. The descriptions are arranged in alphabetical order.

- CHKPT  
 Number of checkpoint calls issued by this transaction.
- CMD  
 Number of command calls issued by this program since PSB scheduling.
- GET CMD  
 Number of GET command calls issued by this program since PSB scheduling.
- MSG GN  
 Number of MESSAGE GET NEXT calls issued by this program since PSB scheduling. (This field does not apply to DBCTL threads, region types DBT and ODB.)
- MSG GU  
 Number of MESSAGE GET UNIQUE calls issued by this program since PSB scheduling. (This field does not apply to DBCTL threads, region types DBT and ODB.)
- MSG ISRT  
 Number of MESSAGE INSERT calls issued by this program since PSB scheduling. For MDPs, this value does not include I/O PCB ISRTs. (This field does not apply to DBCTL threads, region types DBT and ODB.)
- MSG OTHER  
 Number of DC DL/I calls other than GU, GN, CHKPT, PURG, and ISRT. The number includes system service calls, such as SETS and ROLS, and AO application calls, such as CMD and GCMD. (This field does not apply to DBCTL threads, region types DBT and ODB.)

## MSG PURG

Number of MESSAGE PURGE calls issued by this program since PSB scheduling.  
(This field does not apply to DBCTL threads, region types DBT and ODB.)

## PROCLIM

Processing limit count (PROCLIM). Number of transactions that can be processed by this program in one scheduling. If there is no processing limit, this field is blank.  
(This field does not apply to DBCTL threads, region types DBT and ODB.)

<u>Area 3</u>					
-----DL/I CALL ACTIVITY-----				DL/I TOT. .	10
DB GU. . . . .	1	DB GNP. . . .	0	TRAN PROC.	0
DB GHU. . . .	0	DB GHNP. . .	0	DB ISRT. . .	0
DB GN. . . . .	9	DEDB. . . . .	0	DB REPL. . .	0
DB GHN. . . .	0	MSDB. . . . .	0	DB DLET. . .	0

This area shows the amount of DL/I call activity incurred by the transaction currently processing. The descriptions are arranged in alphabetical order.

**Note:** If the TRAN PROC value is greater than 1, the database (DB) counts are cumulative since the start of this PSB scheduling.

## DB DLET

Number of DELETE calls issued by this program against full function databases since PSB scheduling.

## DB GHN

Number of GET HOLD NEXT calls issued by this program against full function databases since PSB scheduling.

## DB GHNP

Number of GET HOLD NEXT within PARENT calls issued by this program against full function databases since PSB scheduling.

## DB GHU

Number of GET HOLD UNIQUE calls issued by this program against full function databases since PSB scheduling.

## DB GN

Number of GET NEXT calls issued by this program against full function databases since PSB scheduling.

## DB GNP

Number of GET NEXT within PARENT calls issued by this program against full function databases since PSB scheduling.

## DB GU

Number of GET UNIQUE calls issued by this program against full function databases since PSB scheduling.

## DB ISRT

Number of INSERT calls issued by this program against full function databases since PSB scheduling.

**DB REPL**

Number of REPLACE calls issued by this program against full function databases since PSB scheduling.

**DL/I TOT**

Total number of DL/I database calls issued by this program since PSB scheduling.

**DEDB**

Total calls issued against DEDB databases since PSB scheduling.

**MSDB**

Total calls issued against MSDB databases since PSB scheduling. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**TRAN PROC**

Number of transaction code messages this program processed. If this value is greater than 1, the database counts are cumulative since the start of this PSB scheduling.

<u>Area 4</u>					
-----OTHER CALL ACTIVITY-----					
CMD. ....	1	SET0. ....	0	APSB. ....	
GCMD. ....	2	SETS. ....	3	DPSB. ....	0
ICMD. ....	0	SETU. ....	0	INIT. ....	0
RCMD. ....	0	ROLB. ....	0	INQY. ....	
GMSG. ....	0	ROLS. ....	0	AUTH. ....	0
CHNG. ....	0	XRST. ....	0	DB DEQ. ...	0

This area shows the amount of IMS call activity, other than the calls shown by the “DL/I CALL ACTIVITY” section, incurred by the transaction currently processing. The descriptions are arranged in alphabetical order.

**APSB**

Number of ALLOCATE PSB calls issued by this program.

**AUTH**

Number of AUTHORIZATION calls issued by this program since PSB scheduling.

**CHNG**

Number of CHANGE DC calls issued by this program since PSB scheduling. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**CMD**

Number of AO application COMMAND calls issued by this program since PSB scheduling.

**DB DEQ**

Number of DATABASE DEQUEUE calls issued by this program since PSB scheduling.

**DPSB**

Number of DEALLOCATE PSB calls issued by this program.

GCMD	Number of AO application GET COMMAND calls issued by this program since PSB scheduling.
GMSG	Number of AO application GET MESSAGE calls issued by this program since PSB scheduling.
ICMD	Number of AO application ICMD calls issued by this program since PSB scheduling.
INIT	Number of INIT calls issued by this program since PSB scheduling.
INQY	Number of INQUIRY calls issued by this program since PSB scheduling.
RCMD	Number of RETRIEVE COMMAND calls issued by this program since PSB scheduling.
ROLB	Number of ROLLBACK calls issued by this program since PSB scheduling.
ROLS	Number of ROLLBACK TO SETS/SETU calls issued by this program since PSB scheduling.
SETO	Number of SET OPTIONS calls issued by this program since PSB scheduling.
SETS	Number of SET BACKOUT POINT calls issued by this program since PSB scheduling.
SETU	Number of SET UNCONDITIONAL calls issued by this program since PSB scheduling.
XRST	Number of EXTENDED RESTART calls issued by this program since PSB scheduling.

<u>Area 5</u>					
----- FP ACTIVITY -----					
NBA. ....	10	DATABASE	AREA	RBA	ENQ
OBA. ....	10	CUSTDB	CUSTDB01	0015BA00	SHR
USED. ....	2	CUSTDB	CUSTDB02	00263300	EXE
		CUSTADD	CUSTADD1	00B19300	SHR

This area shows the region Fast Path call activity for the transaction currently processing. The descriptions are arranged in alphabetical order.

NBA

Number of buffers defined for Normal Buffer Allocation.

OBA

Number of buffers defined for Overflow Buffer Allocation.

USED

Total buffers used. If this total is greater than the NBA value, the extra buffers are taken from the overflow buffer allocation.

The following fields display the first 10 Fast Path database locks held by this region. They show the Fast Path database activity that is occurring for this transaction.

AREA

Fast Path database area being accessed by this transaction.

DATABASE

Fast Path database being accessed by this transaction.

ENQ

ENQ type, which can be either SHR (read) or EXE (update).

RBA

IMS resource ID being locked. For Fast Path DEDBs, this value is the high-order three bytes of the control interval RBA.

<u>Area 6</u>					
----- PSB/TRANSACTION -----					
PSB SIZE. .	968	MODE. . . . .	SNGL	CONVERS. . .	NO
PSB WA. . . .	2520	SEGMENT. . .	SNGLSEG	SPA LEN. . .	0
PROCLIM. . .	0	QUEUED. . . .	0	PRIORITY. . .	0
AVG LENG. . .	200			TOT DEQ. . . .	0

This area shows general information about the application program activity and the transaction currently processing. The descriptions are arranged in alphabetical order.

AVG LENG

The average length of all input messages received with this transaction code (as calculated by the IMS queue manager). For MDPs, this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

CONVERS

Indicates whether this transaction is a conversational transaction (YES) or not (NO). It is blank for Fast Path regions.

MODE

Transaction mode, which can be SNGL or MULT. An IMS synchronization point occurs at each request for a new message (SNGL) or only at program termination (MULT). (This field does not apply to DBCTL threads, region types DBT and ODB.)

PRIORITY

Current scheduling priority of this transaction.

#### PROCLIM

Processing limit count (PROCLIM). Number of transactions that can be processed by this program in one scheduling. If there is no processing limit, this field is blank.

#### PSB SIZE

Total space in bytes used by this program in the PSB pool. This may include the size of the PSB (if not resident), the intent list (if not resident), and a copy of the PDIR (PSB directory entry) if needed for parallel scheduling. If the PSB is resident and not parallel-scheduled, this value is zero.

**Note:** This value includes the size of the PSB in the CSA pool.

#### PSB WA

Size of the work area required by this program in the PSB work pool.

#### QUEUED

Number of currently queued input messages for this transaction code. For MDPs (message-driven program), it is the number of messages queued on this region's BALG. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### SEGMENT

SNGLSEG for single segment or MULTSEG for multiple segments. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### SPA LEN

Length of a conversational transaction's SPA (Scratch Pad Area). If the transaction code is not conversational, this field is blank.

#### TOT DEQ

Total number of messages processed for this transaction code since IMS cold start. For MDPs, the number of messages dequeued from this region's BALG.

**Note:** This counter wraps to zero after reaching 32K.

<u>Area 7</u>				
----- PROGRAM ISOLATION ACTIVITY -----				
	TEST	Q COMMAND	SHR/UPD	EXCLUSIVE
ENQ	0	0	10	1
DEQ	0	0	9	1
CURQ	0	0	1	0
WAIT	0	0	0	0

This area shows IMS program isolation (PI) statistics for the transaction currently running.

These statistics give you the amount of segment-locking activity an application is doing. They are not reset until the transaction terminates. If the application processes multiple transactions in one scheduling, these statistics include the activity of transactions processed earlier in the scheduling.

The columns in this area of the display show the IMS request type. The rows show the type of activity. To interpret the data in this area of the display, look at the row to see the type of activity, then look at the column for the request type. For example, Row 1, Column 1 shows the number of TEST ENQs and Row 2, Column 3 shows the number of SHR or UPD DEQs.

The request types are:

#### TEST

Test enqueues are issued by IMS to determine if a resource is currently being held by another task. Test enqueues do not result in a lock being acquired. They may optionally request to wait for a resource to become available before control is returned to them.

#### Q Command

These are segment reservation requests made by the application program using the Q command code in the segment search argument (SSA).

#### SHR/UPD

These are normal enqueue requests to lock a database segment. SHR locks are for reads (with integrity) and UPD locks are for updates, deletes, and inserts.

The statistics for SHR and UPD are not reported separately, because lock requests are combined into one number by IMS.

#### Exclusive

These are exclusive enqueues. They are issued by IMS administrative tasks that are not related to database segment level requests. Examples of these types of requests are data set opens and extends. They also are used for Fast Path databases whenever two or more regions need a particular control interval or the use of the overflow buffers.

The types of activity reported include:

#### ENQ

Number of enqueue requests since transaction code scheduling to acquire a resource lock.

#### DEQ

Number of dequeue requests since transaction code scheduling to release a resource lock.

#### CURQ

Number of current enqueue requests calculated by subtracting the number of dequeues from the number of enqueues. This only approximates the number of currently held enqueues, because:

- Some dequeue requests only demote a lock from UPD (update) to SHR (share). The request is counted as a dequeue, but the application still holds a lock.
- At synchronization point time, all enqueues are released and are not counted as dequeues. These statistics are not reset until the application program terminates. Thus, the CURQ field may be too high if multiple transactions are processed in the same scheduling.
- The exclusive dequeue count can be larger than the exclusive enqueue count, because the dequeue count at synchronization point may include resources that were not counted as enqueues.

#### WAIT

Number of times the application had to wait since transaction code scheduling for an enqueue to be granted.

### Tuning Tip

To identify an application doing a database scan, look for increasing counts in the ENQ and DEQ rows and no change in count in the CURQ row for the SHR/UPD column.

### Area 8

```

-----SYSTEM ACTIVITY-----RGN ELAPSED 01:30:04
EXEC IN... DLI-SAS          TCB CPU... 1,740      DOMAIN.... 1
ASID..... 11              SRB CPU... 234        PERF GR... 13
PRIORITY.. 238            CPU SRV... 10968       PERF PD... 1
SWAP..... NON-SWAP        IO SRV... 3555
                                MSO SRV... 6

```

This area shows general information about the region in relation to the operating system. The descriptions are arranged in alphabetical order.

#### ASID

OS/390 address space identification number.

#### CPU SRV

Total CPU service units for this address space.

#### DOMAIN

Address space domain used by SRM. N/A is shown if WLM is in goal mode.

#### EXEC IN

Execution is in the control or dependent region:

ALC	Currently executing in a TCB for IMS internal use
ALM	Currently executing in a selective dispatching TCB (application program maintaining LU 6.2 conversation for a long time)
BMP	Currently executing under a batch message processing region TCB
CTL	Currently executing in the control region TCB
CTX	Currently executing under a control region auxiliary TCB
DBT	Currently executing under a DBCTL (CICS attachment) region TCB
DCC	Currently executing under a DCCTL management TCB
DLG	Currently executing in the DASD log TCB
DLI	Currently executing in a TCB in the DL/I address space
DRA	Currently executing under a database resource adapter thread (DBCTL connection) TCB
DRC	Currently executing in a database recovery processing TCB
DTT	Currently executing in a TCB for IMS internal use
DYA	Currently executing in a dynamic allocation TCB
DYD	Currently executing in a TCB for IMS internal use

DYS	Currently executing in a dynamic control services TCB (dynamic allocation service in the DL/I subordinate address space)
ESI	Currently executing in an external subsystem interface TCB
ESS	Currently executing an external subsystem request TCB
IFP	Currently executing under a Fast Path region TCB
JBP	Currently executing under a Java batch message processing TCB
JMP	Currently executing under a Java message processing TCB
LRD	Currently executing in a TCB for IMS internal use
LRM	Currently executing in a TCB for IMS internal use
LSD	Currently executing in a local storage management TCB
LUM	Currently executing in an APPC request management TCB
MOD	Currently executing in an IMS control, MODIFY TCB
MPP	Currently executing under a message processing region TCB
ODM	Currently executing under the ODBA syncpoint mother TCB
ODS	Currently executing under an ODBA syncpoint daughter TCB
OII	Currently executing under the OTMA initialization TCB
OIC	Currently executing under the OTMA control TCB
OIM	Currently executing under an OTMA member TCB
RCF	Currently executing under the allocate RCF TCB
RDS	Currently executing in a restart data set processing TCB
RLM	Currently executing in an IRLM request processing TCB
RST	Currently executing in a restart processing TCB
RWD	Currently executing in the online recovery service daughter TCB
RWM	Currently executing in the online recovery service main TCB
SQ1	Currently executing in the shared queues SQ1 TCB
SQ2	Currently executing in the shared queues SQ2 TCB
STC	Currently executing in a storage compression TCB
STM	Currently executing in a storage management TCB (job step TCB)
TMC	Currently executing in a time controlled option (TCO) TCB
TRA	Currently executing in an IMS trace processing TCB
XCF	Currently executing in an Extended Communication Facility (LU 6.2) TCB
XFP	Currently executing in a Fast Path common services TCB
xx=?	Displayed when an unknown task ID (TCB) is encountered The unknown task ID replaces xx (contact BMC Software Customer Support).

#### IO SRV

Total I/O service units for this address space.

#### MSO SRV

Total main storage service units for this address space.

#### PERF GR

Address space performance group used by SRM. N/A is shown if WLM is in goal mode.

#### PERF PD

Address space current performance period used by SRM. N/A is shown if WLM is in goal mode.

#### PRIORITY

Current priority of this address space on the dispatch queue. N/A is shown if WLM is in goal mode.

#### RGN ELAPSED

Elapsed time the region has been active (current time - region start time) expressed in hours, minutes, and seconds (hh:mm:ss).

#### SRB CPU

Total region SRB CPU time expressed in seconds.

#### SWAP

Current swapping status of the address space:

IN	Swapped in
NON-SWAP	Nonswappable
OUT	Swapped out

#### TCB CPU

Total region TCB CPU time expressed in seconds.

<u>Area 9</u>					
-----PAGING ACTIVITY-----					
WRK IN...	0	INT PGI...	0	VIO PGI...	0
WRK OUT...	0	INT PGO...	0	VIO PGO...	0
FRAMES...	0	INT RCL...	0	VIO RCL...	0
SLOTS...	0	SWAPCNT...	1	SLT VIO...	0
COM PGI...	5	COM RCL...	0	STOLEN...	5434
***** END OF DATA *****					

This area shows a summary of paging activity in the region. The descriptions are arranged in alphabetical order.

Paging counts refer to the current SRM interval (time since last swap-in). These count fields are blank when the corresponding address space is swapped out. When the region is nonswappable, the counts are totals accumulated since the region started.

#### COM PGI

Number of page-ins in the common area (CSA/LPA) for this address space during this interval.

COM RCL	Number of page reclaims in the common area (CSA/LPA) for this address space during this interval.
FRAMES	Number of page frames of real storage currently allocated to this address space.
INT PGI	Number of page-ins during this interval.
INT PGO	Number of page-outs during this interval.
INT RCL	Number of page reclaims during this interval.
SLOTS	Number of pages (slots) of direct access storage on the paging device that is allocated to this address space. The number includes VIO and non-VIO slots.
SLT VIO	Number of pages (slots) of direct access storage on the paging devices that are allocated to this address space for VIO.
STOLEN	Number of pages stolen for this address space during this interval.
SWAPCNT	Number of times this address space has been swapped out. Even though the regions are marked nonswappable, they may swap during initialization. After this, the count remains the same.
VIO PGI	Number of page-ins for VIO during this interval.
VIO PGO	Number of page-outs for VIO during this interval.
VIO RCL	Number of VIO page reclaims during this interval.
WRK IN	Number of page frames last swapped in.
WRK OUT	Number of page frames last swapped out.

## REGNS - IMS Regions (Event Collector Data)

### REGNS

#### Summary View

(Default view  
when REGNS is  
requested)

```
BMC SOFTWARE ----- IMS REGIONS ----- PERFORMANCE MGMT
SERV ==> REGNS          INPUT  10:38:57  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==> SO=NA, V=SM                      ROW   1 OF   4  SCROLL=> CSR
VIEW:   SM, MSG, DLI, DB2                EXPAND: FSEL(+), LINESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW                      *EVENT COLLECTOR*

  ID NAME      TYP STATUS   PSB      TRANCODE  USR/LTRM  DLI   SQL LOCKS TRN-ELAP
  -----
    1 CICS3302 DBT IDLE
    2 CICS3302 DBT SCHEDULE STBCUSR  DLZZ              0    0    0    450ms
    3 CICS3302 DBT IDLE
    4 IMSM17X  MPP IDLE
***** END OF DATA *****
```

### REGNS

#### Message View

```
BMC SOFTWARE ----- IMS REGIONS ----- PERFORMANCE MGMT
SERV ==> REGNS          INPUT  10:40:42  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==> SO=NA, V=MSG                      ROW   1 OF   4  SCROLL=> CSR
VIEW:   SM, MSG, DLI, DB2                EXPAND: FSEL(+), LINESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW                      *EVENT COLLECTOR*
  NAME      TYP CL1 CL2 CL3 CL4 TRANCODE  PRLIM QUEUED M-DEQ M-GN M-PRG M-ISR
  -----
  CICS3302 DBT
  CICS3302 DBT                      DLZZ              0    0    0    0    0
  CICS3302 DBT
  IMSM17X  MPP    1    2    3    4
***** END OF DATA *****
```

### REGNS

#### DLI View

```
BMC SOFTWARE ----- IMS REGIONS ----- PERFORMANCE MGMT
SERV ==> REGNS          INPUT  10:40:42  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==> SO=NA, V=DLI                      ROW   1 OF   4  SCROLL=> CSR
VIEW:   SM, MSG, DLI, DB2                EXPAND: FSEL(+), LINESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW                      *EVENT COLLECTOR*
  NAME      TRANCODE #DB  GU   GN  GHU  GHN REPL ISRT DLET  TOT CHKP I/O
  -----
  MPPRGN2  SM4          2    4    2    0    0    0    0    0    6    0    0
  MPPRGN3  LMZ200XX    1    1    2    0    0    0    0    0    3    0    1
  CICSRCN1 DZ33        1    5    4    0    0    0    0    0    9    0    2
  MPPRGN9  APPCPGM1    2    2    1    0    0    0    0    0    3    0    1
***** END OF DATA *****
```

## REGNS DB2 View

BMC SOFTWARE-----				IMS REGIONS				-----PERFORMANCE MGMT			
SERV ==>	REGNS		INPUT	10: 40: 42	INTVL=>	3	LOG=>	N	TGT==>	IMSA	
PARM ==>	SO=NA, V=DB2				ROW	1	OF	4	SCROLL=>	CSR	
VIEW:	SM, MSG, DLI, DB2				EXPAND:	FSEL(+),	LINESEL(DREGN),	MON(RGN)			
PF10/11	FOR PREV/NEXT VIEW								*EVENT COLLECTOR*		
NAME	TRANCODE	FETCH	OPEN	ISRT2	DLET2	UPDT2	DDL	DYN	CNTL	OTHER	TOT2
-----	++++++	-----	-----	-----	-----	-----	-----	-----	-----	-----	++++
MPPRGN2	SM4	1	0	0	0	0	0	0	0	0	1
MPPRGN3	LMZ200XX	4	1	0	0	0	0	0	0	0	5
CICSRGN1	DZ33										
MPPRGN9	APPCPGM1	0	0	0	0	0	0	0	0	0	
***** END OF DATA *****											

### Description:

This value is the Event Collector version of the REGNS display. If the Event Collector is not active, only the data shown by “REGNS - IMS Regions (No Event Collector Data)” on page 187 is displayed.

This version of REGNS takes advantage of additional data collected by the Event Collector. It shows IMS dependent region activity and the work currently being performed by IMS for the transaction currently processing. REGNS arranges this information by views. For example, there is a DL/I view that shows number of databases accessed, counts for number of checkpoints, and the number of I/Os incurred by a transaction.

The following views of dependent region activity for a transaction are available when the Event Collector is active:

SM (summary)	Shows status information about your active regions and the transactions that are currently executing. This is the default display.
MSG (message)	Shows information related to scheduling and the message queue.
DLI	Shows information by region about transaction DL/I database activity.
DB2	Shows information about a transaction's DB2 activity.

When the Event Collector is active, \*EVENT COLLECTOR\* is shown in the fifth line of the REGNS display.

When the Event Collector is not active, REGNS is limited to the amount of data it can display as shown by “REGNS - IMS Regions (No Event Collector Data)” on page 187. Also, REGNS without the Event Collector shows the data for the duration of a PSB scheduling rather than for the transactions currently processing.

**Color:**

If you have a color monitor:

Blue	Indicates the current view.
Red	Highlights regions (rows) that are in a wait state.
Turquoise	Indicates normal values.
Yellow	Indicates an error message.
Pink	Indicates a WFI or pseudo-WFI transaction is waiting for input. This affects the STATUS, PSB, and TRANCODE columns.

**Select Code:**

REGNS

**Parameter:**

All the REGNS parameters, except for SORT and VIEW, act as filters that restrict the information shown according to the criteria specified by the parameter(s). SORT sorts the columns in the display by specified characters. VIEW shows the requested view of REGNS data. The REGNS parameters can be used as follows:

- The filtering parameters apply across all views. All views are restricted by the specified parameter.

For example, if you enter:

TYPE=DBT

in the DLI view, only the rows for DBCTL threads are displayed. A filtering parameter can be used with any view even though that view does not display the applicable field.

**Tuning Tip**

Use the NI or NOTIDLE parameter to see only those regions that are currently processing transactions.

- Multiple parameters must be separated by commas.
- A blank indicates the end of a parameter string.
- Multiple resources with similar names can be requested by using an \* character as a generic qualifier and a + character as a positional qualifier. The positional qualifier is repeated for every character to be replaced. The generic qualifier replaces all characters at the end of the parameter.

For example, a parameter of NAME=MP\* shows all region names that start with MP and have any character in the rest of the name.

- If multiple filtering parameters are entered, the regions displayed must meet all the restrictions.
- If one parameter invalidates another, an error message is issued without further processing.

The following parameter descriptions are arranged alphabetically. The descriptions begin with parameters that start with a special character. Parameters containing a numeric character are first in their alphabetical group.

#D|#DB<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents the number of different DL/I databases accessed. It displays in each view only those regions with transactions that have a DL/I database access count less than (<), greater than (>), or equal to (=) the specified number.

AC|ACTCLASS=nnn|(nnn,...)

Where nnn represents a transaction class number. It displays in each view only those regions that are currently running a transaction in the specified class(es). A maximum of four classes can be specified.

AGN=xxxxxxxx

Where xxxxxxxx represents the name of an application group. It displays in each view only those regions with a name that matches the specified application group name. The application group name for a region is shown in the AGN field of the DREGN display.

CK|CHKP<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a checkpoint count. It displays in each view only those regions with transactions that have a total checkpoint call count less than (<), greater than (>), or equal to (=) the specified number.

C1|CL1=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the first class equal to the specified number.

C2|CL2=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the second class equal to the specified number.

C3|CL3=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the third class equal to the specified number.

C4|CL4=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the fourth class equal to the specified number.

CL|CLASS=nnn|(nnn,...)

Where nnn represents a transaction class number. It displays in each view only those regions that can process this class(es) of transaction. A maximum of four classes can be specified.

CN|CNTL<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DB2 control-type SQL call count (GRANT, for example). It displays in each view only those regions with transactions that have a total DB2 control SQL count less than (<), greater than (>), or equal to (=) the specified number.

DD|DDL<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DB2 Data Definition Language (DDL) count. It displays in each view only those regions with transactions that have a total DB2 DDL count less than (<), greater than (>), or equal to (=) the specified number.

DT|DLET<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DL/I database delete count. It displays in each view only those regions with transactions that have a database delete count less than (<), greater than (>), or equal to (=) the specified number.

D2|DLET2<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DB2 database delete count. It displays in each view only those regions with transactions that have a total DB2 delete count less than (<), greater than (>), or equal to (=) the specified number.

DL|DLI<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnn represents the total DL/I call count. It displays in each view only those regions currently executing transactions with a total DL/I call count less than (<), greater than (>), or equal to (=) the specified number.

DY|DYN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DB2 dynamic call count. It displays in each view only those regions with transactions that have a total DB2 dynamic call count less than (<), greater than (>), or equal to (=) the specified number.

GN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a GET NEXT and GET NEXT within PARENT count. It displays in each view only those regions with transactions that have a database GN and GNP count less than (<), greater than (>), or equal to (=) the specified number.

GU<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a GET UNIQUE count. It displays in each view only those regions with transactions that have a database GU count less than (<), greater than (>), or equal to (=) the specified number.

HN|GHN<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a GET HOLD NEXT and GET HOLD NEXT within PARENT count. It displays in each view only those regions with transactions that have a database GHN and GHNP count less than (<), greater than (>), or equal to (=) the specified number.

HU|GHU<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a GET HOLD UNIQUE count. It displays in each view only those regions with transactions that have a database GHU count less than (<), greater than (>), or equal to (=) the specified number.

IO<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a DL/I I/O count. It displays in each view only those regions with transactions that have a total DI/I I/O count less than (<), greater than (>), or equal to (=) the specified number.

ID<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a region number. It displays in each view only those regions with a region number less than (<), greater than (>), or equal to (=) the specified number.

IS|ISRT<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents an INSERT count. It displays in each view only those regions with transactions that have a database ISRT count less than (<), greater than (>), or equal to (=) the specified number.

I2|ISRT2<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a DB2 INSERT count. It displays in each view only those regions with transactions that have a total DB2 ISRT count less than (<), greater than (>), or equal to (=) the specified number.

LO|LOCKS<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a PI or IRLM enqueue count. It displays in each view only those regions currently executing transactions with either outstanding PI enqueues or IRLM locks (if IMS IRLM is active) less than (<), greater than (>), or equal to (=) the specified number.

LT=xxxxxxx

Where xxxxxxxx represents an LTERM name. It displays in each view only those regions processing transactions originating from an LTERM that matches the specified LTERM name. Multiple regions can be displayed by using an \* or + qualifier; for example, LT=SNL++060.

MD|M-DEQ<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message dequeue count. It displays in each view only those regions with transactions that have a message dequeue count less than (<), greater than (>), or equal to (=) the specified number.

MG|M-GN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message GET NEXT count. It displays in each view only those regions with transactions with message GN counts less than (<), greater than (>), or equal to (=) the specified number.

MI|M-ISR<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message INSERT count. It displays in each view only those regions with transactions that have a message ISRT count less than (<), greater than (>), or equal to (=) the specified number.

MP|M-PRG<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message PURGE count. It displays in each view only those regions with transactions that have a message PURG count less than (<), greater than (>), or equal to (=) the specified number.

NA|NAME=xxxxxxx

Where xxxxxxxx represents a region name. It displays in each view only those regions with a name that matches the specified region name. Multiple regions can be displayed by using an \* or + qualifier; for example, NAME=\*RGN1.

NI|NOTIDLE

It displays in each view only those regions that are not idle (active).

OP|OPEN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DB2 OPEN cursor count. It displays in each view only those regions with transactions that have a total DB2 OPEN cursor count less than (<), greater than (>), or equal to (=) the specified number.

OT|OTHER<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a count of other DB2 SQL calls. It displays in each view only those regions with transactions that have a count for DB2 SQL calls other than those defined for this display that is less than (<), greater than (>), or equal to (=) the specified number.

PL|PRLIM<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a region processing limit count. It displays in each view only those regions with transactions that have a processing limit count less than (<), greater than (>), or equal to (=) the specified number.

PS|PSB=xxxxxxx

Where xxxxxxxx represents a PSB name. It displays in each view only those regions with PSB names that match the specified name. Multiple regions can be displayed by using an \* or + qualifier; for example, PSB=SM++0+00.

QU|QUEUED<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a transaction message input queue count. It displays in each view only those regions with transactions that have a message input queue count that is less than (<), greater than (>), or equal to (=) the specified number.

RE|REPL<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a database REPLACE count. It displays in each view only those regions with transactions that have a database REPL count less than (<), greater than (>), or equal to (=) the specified number.

SE|SEL<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a count of DB2 SQL SELECT and FETCH calls. It displays in each view only those regions with transactions that have a total SELECT and FETCH call count less than (<), greater than (>), or equal to (=) the specified number.

SO|SORT=cc

Where cc represents any of the following two-character view columns. The display is sorted by NAME by default. The sort sequence is ascending for alphanumeric characters and descending for numeric characters except for ID. The numeric region ID is sorted in ascending sequence.

**Tip**

An easy way to sort is to move the cursor to the column heading and press ENTER.

The following SORT parameter descriptions are arranged alphabetically. SORT keywords that start with a special character are described first. Keywords containing a numeric character are first in their alphabetical group. All views are sorted as requested by the specified parameter.

Any of these keywords can be used in any view.

#D	Sorts the list by #DB (descending).
CK	Sorts the list by CHKPT (descending).
C1	Sorts the list by CL1 (descending).
C2	Sorts the list by CL2 (descending).
C3	Sorts the list by CL3 (descending).
C4	Sorts the list by CL4 (descending).
CN	Sorts the list by CNTL (descending).
D2	Sorts the list by DLET2 (descending).
DD	Sorts the list by DDL (descending).
DL	Sorts the list by DLI (descending).
DT	Sorts the list by DLET (DL/I database; descending).
DY	Sorts the list by DYN (descending).
GN	Sorts the list by GN (DL/I database; descending).
GU	Sorts the list by GU (DL/I database; descending).
HN	Sorts the list by GHN (DL/I database; descending).
HU	Sorts the list by GHU (DL/I database; descending).
I2	Sorts the list by ISRT2 (descending).
ID	Sorts the list by ID (numeric region ID; ascending).
IO	Sorts the list by I/O (descending).
IS	Sorts the list by ISRT (DL/I database; descending).
LO	Sorts the list by LOCKS (descending).
LT	Sorts the list by USR/LTRM (ascending).
MD	Sorts the list by M-DEQ (descending).
MG	Sorts the list by M-GN (descending).
MI	Sorts the list by M-ISRT (descending).
MP	Sorts the list by M-PRG (descending).
NA	Sorts the list by NAME (default; ascending).
OP	Sorts the list by OPEN (descending).
OT	Sorts the list by OTHER (descending).
PL	Sorts the list by PRLIM (descending).
PS	Sorts the list by PSB name (ascending).
QU	Sorts the list by QUEUED (descending).
RE	Sorts the list by REPL (DL/I database; descending).
SE	Sorts the list by SEL/FETCH (descending).
SQ	Sorts the list by SQL (descending).
ST	Sorts the list by STATUS (ascending).

T2 Sorts the list by TOT2 (descending).  
 TE Sorts the list by TRN-ELAP (descending).  
 TO Sorts the list by TOT (DL/I database; descending).  
 TR Sorts the list by TRANCODE (ascending).  
 TY Sorts the list by TYP (ascending).  
 U2 Sorts the list by UPDT2 (descending).  
 US Sorts the list by USR/LTRM (ascending).

SQL|SQL<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents an SQL call count. It displays in each view only those regions currently executing transactions with SQL call counts less than (<), greater than (>), or equal to (=) the specified number. Specify SQL>0 for any transaction that accesses DB2.

ST|STATUS=

It displays in each view only those regions that meet the specified criteria. Multiple regions can be displayed by using an \* or + qualifier. For example, you can select all:

- Active regions by specifying ST=A\*
- Idle regions by specifying ST=I D\*
- Waiting regions by specifying ST=W\*

Any of the region statuses described on page 119 can be specified with the STATUS parameter.

TO|TOT<nnn|>nnn|=nnn

Where nnn represents a total DL/I database call count. It displays in each view only those regions with transactions that have a total DL/I database call count less than (<), greater than (>), or equal to (=) the specified number.

T2|TOT2<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a total DB2 SQL call count. It displays regions with transactions that have a total DB2 SQL count less than (<), greater than (>), or equal to (=) the specified number.

TR|TRANCODE=xxxxxxxx

Where xxxxxxxx represents a transaction code. It displays in each view only those regions that have transaction codes that match the specified code. Multiple regions can be displayed by using an \* or + qualifier; for example, TR=+MZ200XX.

TE|TRN-EL<nnn|>nnn|=nnn

Where nnn represents elapsed time in seconds. It displays regions currently executing transactions that have an elapsed time less than (<), greater than (>), or equal to (=) the specified time in seconds.

TY|TYP=xxx

Where xxx represents a region type. Multiple regions can be displayed by using an \* or + qualifier; for example, TYP=M\*.

The following type descriptions are arranged alphabetically:

BMI	Displays batch message processing regions currently executing an implicit APPC/IMS transaction.
BMO	Displays batch message processing regions currently executing an OTMA transaction.
BMP	Displays batch message processing regions.
BMW	Displays wait-for-input BMP.
DBT	Displays DBCTL CICS threads.
FPU	Displays Fast Path utility regions.
JBP	Displays Java batch message processing regions.
JMI	Displays Java message processing regions currently executing an implicit APPC/IMS transaction.
JMO	Displays Java message processing regions currently executing an OTMA transaction.
JMP	Displays Java message processing regions.
JMW	Displays wait-for-input JMP.
MDP	Displays message-driven Fast Path regions.
MPI	Displays message processing regions currently executing an implicit APPC/IMS transaction.
MPO	Displays message processing regions currently executing an OTMA transaction.
MPP	Displays message processing regions.
MPW	Displays wait-for-input MPP.
NDP	Displays non-message-driven Fast Path regions.
ODB	Displays DBCTL ODBA threads.
TPI	Displays message processing regions currently executing an explicit CPI-C program.

U2|UPDT2<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a DB2 update count. It displays in each view only those regions with transactions that have a total DB2 update count less than (<), greater than (>), or equal to (=) the specified number.

US|USR=xxxxxxxx

Where xxxxxxxx represents a user ID. It displays in each view only those regions with user IDs that match the specified ID. Multiple regions can be displayed by using an \* or + qualifier; for example, USR=CICS\*.

V|VIEW=SM|MSG|DLI|DB2

Displays the specified view:

SM	Displays the REGNS summary status view (default).
MSG	Displays the IMS/DC message view.
DLI	Displays the DL/I database view.
DB2	Displays the DB2 database activity view.

**View Selection:**

You can:

- Enter the VIEW parameter in the PARM field.
- Move the cursor to the view you want in the VIEW line and press ENTER.
- Use one of the following PF keys:

PF10      Displays the previous view.

PF11      Displays the next view.

**Expand:**

The REGNS display can be EXPANDED by using the following fields as indicated:

FSEL(+)

Indicates that at least one column has cursor-sensitive fields. A string of + characters underneath a column name means this column contains cursor-sensitive fields. Moving the cursor to a field in that column and pressing the ENTER key invokes a related display for that field. For example, if the cursor is at a field value under the DLI column, pressing the ENTER key displays the DLIST service with the correct region number as the parameter. Columns with cursor-sensitive fields include:

DLI

The DLIST service showing DL/I call details can be displayed with the region number as a parameter.

TRANCODE

The TRANQ service showing transaction status details can be displayed with the transaction code as a parameter.

You can also hyperlink from this field to MAINVIEW for CICS as described on page 176.

SQL

You can hyperlink from this field to MAINVIEW for DB2 as described on page 176.

TOT2

You can hyperlink from this field to MAINVIEW for DB2 as described on page 176.

## LOCKS

If IMS IRLM is active, the LUSRD service showing IRLM lock user details can be displayed with the region number as a parameter. If IRLM is not active, the PI service showing program isolation information can be displayed with the region number as a parameter.

## QUEUED

The CLASQ service showing transaction queuing and processing can be displayed with the region number as a parameter.

## TOT

The DLIST service showing DL/I call details can be displayed with the region number as a parameter.

When a display service is cursor-selected from a REGNS field, that service shows system data at the time it is invoked. It is not synchronized with the REGNS display.

Use the END PF key to return to the REGNS display.

## LINESEL(DREGN)

Move the cursor to a region in one of the rows of a REGNS view and press ENTER to invoke the DREGN service. DREGN shows region details for the selected region.

Selecting DREGN with the cursor from the EXPAND line invokes the DREGN service for the region in the first row of the REGNS display.

## MON(RGN)

Move the cursor to this field in the EXPAND line to display the Active Timer Request service for all active region monitors.

### **Hyperlink:**

If MAINVIEW for CICS is installed and a transaction is a CICS transaction, you can link to the TASKXPND service in MAINVIEW for CICS by using the TRANCODE column in any view to cursor-select that transaction.

If MAINVIEW for DB2 is installed and a transaction has an active DB2 thread, you can link to the DUSER service in MAINVIEW for DB2 by using the SQL column in the summary view or the TOT2 column in the DB2 view to cursor-select that transaction.

**Sorting:**

The display list can be sorted by:

- Using the SORT parameter
- Positioning the cursor with the TAB key to the column heading to be sorted and pressing ENTER

This overrides any SORT parameter in the PARM field and primes that field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by NAME by default. Alphanumeric fields and the ID field are sorted in ascending order. Numeric fields are sorted in descending order. Fields with no values are sorted to the bottom of the list.

**Scrolling:**

The display is scrollable.

**Field Descriptions:**

Each of the fields is shown and described below by view.

<u>Summary View</u>									
BMC SOFTWARE -----					IMS REGIONS		----- PERFORMANCE MGMT		
SERV ==>	REGNS		INPUT	10:38:57	INTVL=>	3	LOG=>	N	TGT==> IMSA
PARM ==>	SO=NA, V=SM				ROW	1 OF	4	SCROLL=>	CSR
VIEW:	SM, MSG, DLI, DB2				EXPAND:	FSEL(+),	LINESEL(DREGN),	MON(RGN)	
PF10/11 FOR PREV/NEXT VIEW					*EVENT COLLECTOR*				
ID NAME	TYP	STATUS	PSB	TRANCODE	USR/LTRM	DLI	SQL	LOCKS	TRN-ELAP
-----					++++++	-----	++++	++++	++++
1	CICS3302	DBT	IDLE						
2	CICS3302	DBT	SCHEDULE	STBCUSR	DLZZ		0	0	0
3	CICS3302	DBT	IDLE						450ms
4	IMSM17X	MPP	IDLE						
***** END OF DATA *****									

This view shows the status of active regions and transactions. The descriptions are arranged in alphabetical order.

**DLI**

Total DL/I database call count for this transaction.

You can expand from this column to the DLIST service.

**ID**

Region identifier assigned to this region by IMS.

**LOCKS**

Either total outstanding PI enqueue counts or IRLM locks if IMS IRLM is active. An \* character following the count indicates that this transaction is waiting for a PI or IRLM lock if IMS IRLM is active.

You can expand from this column to the LUSRD service if IRLM is active. Otherwise, you can expand to the PI service.

**NAME**

Job name or started task procedure name of the region.

PSB	PSB name of the application program currently being processed.
SQL	Total SQL count. You can hyperlink from this column to the DUSER service in MAINVIEW for DB2 if it is installed and there is an active DB2 thread. (This field does not apply to DBCTL threads, region types DBT and ODB.)
STATUS	Region status:
ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (the Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on).  The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.

WAIT-AOI	Region waiting for an AO message.  Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch).  The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.

WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### TRANCODE

Name of transaction currently being processed.

For an IMS transaction, you can expand from this column to the TRANQ service. For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.

#### TYP

Type of region processing a transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.

BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

#### USR/LTRM

ID of the user who submitted this transaction or LTERM where transaction was submitted. An \* character indicates a terminal ID was used instead of a user ID.

Message View									
BMC SOFTWARE-----					IMS REGIONS			-----PERFORMANCE MGMT	
SERV ==>	REGNS	INPUT			10: 40: 42	INTVL=>	3	LOG=>	N TGT==> IMSA
PARM ==>	S0=NA, V=MSG					ROW	1 OF	4	SCROLL=> CSR
VIEW:	SM, MSG, DLI, DB2	EXPAND:			FSEL(+),	LINESEL(DREGN),	MON(RGN)		
PF10/11	FOR PREV/NEXT VIEW								*EVENT COLLECTOR*
NAME	TYP	CL1	CL2	CL3	CL4	TRANCODE	PRLIM	QUEUED	M-DEQ M-GN M-PRG M-ISR
-----	-----	-----	-----	-----	-----	++++++	-----	++++++	-----
CICS3302	DBT								
CICS3302	DBT				DLZZ		0	0	0 0 0 0
CICS3302	DBT								
IMSM17X	MPP	1	2	3	4				
***** END OF DATA *****									

This view shows transaction activity in each active IMS dependent region. The descriptions are arranged in alphabetical order.

For the following fields, CL1 through CL4, the currently active class is shown by a > character preceding the class:

#### CL1

First class this MPP can process. (This field does not apply for a TYP of DBT and ODB and for all Fast Path types, including MDP and NDP.)

#### CL2

Second class this MPP can process. (This field does not apply for a TYP of DBT and ODB and for all Fast Path types, including MDP and NDP.)

#### CL3

Third class this MPP can process. (This field does not apply for a TYP of DBT and ODB and for all Fast Path types, including MDP and NDP.)

#### CL4

Fourth class this MPP can process. (This field does not apply for a TYP of DBT and ODB and for all Fast Path types, including MDP and NDP.)

#### M-DEQ

Number of messages successfully processed by the application program in this scheduling. For MDPs (message-driven program), this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### M-GN

Number of MESSAGE GET NEXT calls issued. For MDPs, this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### M-ISR

Number of message insert calls issued. For MDPs, this value does not include I/O PCB ISRTs. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### M-PRG

Number of message purge calls issued. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**NAME**

Job name or started task procedure name of the region.

**PRLIM**

Processing limit count of the transaction. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**QUEUED**

Number of currently queued input messages with this transaction code. For MDPs (message-driven), the number of messages queued on this region's BALG. You can expand from this column to the CLASQ service. (This field does not apply to DBCTL threads, region types DBT and ODB.)

**TRANCODE**

Name of transaction currently being processed. For an IMS transaction, you can expand from this column to the TRANQ service. For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.

**TYP**

Type of region processing a transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.

MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

<u>DL/I View</u>												
BMC SOFTWARE-----				IMS REGIONS				-----PERFORMANCE MGMT				
SERV ==>	REGNS		INPUT	10: 40: 42	INTVL=>	3	LOG=>	N	TGT==>	IMSA		
PARM ==>	SO=NA, V=DLI						ROW	1 OF	4	SCROLL=>	CSR	
VIEW:	SM, MSG, DLI, DB2				EXPAND:	FSEL(+),	LINESEL(DREGN),	MON(RGN)				
PF10/11	FOR PREV/NEXT VIEW									*EVENT COLLECTOR*		
NAME	TRANCODE	#DB	GU	GN	GHU	GHN	REPL	ISRT	DLET	TOT	CHKP	I/O
-----	++++++	----	----	----	----	----	----	----	----	++++	----	----
MPPRGN2	SM4	2	4	2	0	0	0	0	0	6	0	0
MPPRGN3	LMZ200XX	1	1	2	0	0	0	0	0	3	0	1
CICSRGN1	DZ33	1	5	4	0	0	0	0	0	9	0	2
MPPRGN9	APPCPGMI	2	2	1	0	0	0	0	0	3	0	1
***** END OF DATA *****												

This view shows transaction DL/I database activity in each active IMS dependent region. The descriptions are arranged in alphabetical order.

#### #DB

Number of DL/I databases accessed so far.

#### CHKPT

Total checkpoint calls (this information is useful for BMP checkpoint frequency estimates) for this transaction.

#### DLET

Total delete calls for this transaction.

#### GHN

Sum of GET HOLD NEXT and GET HOLD NEXT within PARENT calls for this transaction.

#### GHU

Total GET HOLD UNIQUE calls for this transaction.

#### GN

Sum of GET NEXT and GET NEXT within PARENT calls for this transaction.

#### GU

Total database GET UNIQUE calls for this transaction.

#### I/O

Total read and write counts for this transaction.

#### ISRT

Total insert calls for this transaction.

NAME

Job name or started task procedure name of the region.

REPL

Total replace calls for this transaction.

TOT

Total DL/I database calls for this transaction. You can expand from this column to the DLIST service.

TRANCODE

Name of transaction currently being processed. For an IMS transaction, you can expand from this column to the TRANQ service. For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.

<u>DB2 View</u>											
BMC SOFTWARE-----				IMS REGIONS				-----PERFORMANCE MGMT			
SERV ==>	REGNS		INPUT	10: 40: 42	INTVL=>	3	LOG=>	N	TGT==>	IMSA	
PARM ==>	SO=NA, V=DB2				ROW	1	OF	4	SCROLL=>	CSR	
VIEW:	SM, MSG, DLI, DB2				EXPAND:	FSEL(+),	LINESEL(DREGN),	MON(RGN)			
PF10/11	FOR PREV/NEXT VIEW								*EVENT COLLECTOR*		
NAME	TRANCODE	FETCH	OPEN	ISRT2	DLET2	UPDT2	DDL	DYN	CNTL	OTHER	TOT2
-----	++++++	-----	-----	-----	-----	-----	-----	-----	-----	-----	+++++
MPPRGN2	SM4	1	0	0	0	0	0	0	0	0	1
MPPRGN3	LMZ200XX	4	1	0	0	0	0	0	0	0	5
CICSRGN1	DZ33										
MPPRGN9	APPCPGM1	0	0	0	0	0	0	0	0	0	
***** END OF DATA *****											

This view shows DB2 transaction activity. The descriptions are arranged in alphabetical order.

CNTL

Total number of SQL control-type calls, such as REVOKE, issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)

DDL

Total number of Data Definition Language calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)

DLET2

Total number of SQL DELETE calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)

DYN

Total number of dynamic SQL calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)

FETCH	Total number of SQL SELECT and FETCH calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)
ISRT2	Total number of SQL INSERT calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)
NAME	Job name or started task procedure name of the region.
OPEN	Total number of SQL OPEN calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)
OTHER	Total number of other SQL calls issued to DB2 by the transaction currently processing. These are call types that do not fit in any of the other types defined in this display, such as table and security manipulation. (This field does not apply to DBCTL threads, region types DBT and ODB.)
TOT2	Total SQL calls issued to DB2 by the transaction currently processing. You can hyperlink from this column to the DUSER service in MAINVIEW for DB2 if it is installed and there is an active DB2 thread. (This field does not apply to DBCTL threads, region types DBT and ODB.)
TRANCODE	Name of transaction currently being processed. For an IMS transaction, you can expand from this column to the TRANQ service. For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.
UPDT2	Total number of SQL UPDATE calls issued to DB2 by the transaction currently processing. (This field does not apply to DBCTL threads, region types DBT and ODB.)

## REGNS - IMS Regions (No Event Collector Data)

### REGNS

#### Summary View

(Default view  
when REGNS is  
requested)

```
BMC SOFTWARE ----- IMS REGIONS ----- PERFORMANCE MGMT
SERV ==> REGNS          INPUT  10:38:57  INTVL=> 3  LOG=> N  TGT==> IMSA
PARAM ==> SO=NA, V=SM          ROW    1 OF    4  SCROLL=> CSR
VIEW:    SM, MSG, DLI          EXPAND: FSEL(+), LINESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW

  ID NAME      TYP STATUS   PSB      TRANCODE  USR/LTRM   DLI DB2 LOCKS ELAPSED
  -----
    1 CICS3302 DBT IDLE
    2 CICS3302 DBT ACTV-USR STBCUSR  DLZZ              10          1
    3 CICS3302 DBT IDLE      ABEM003T ABMAIL  CIR11          4 YES
***** END OF DATA *****
```

### REGNS

#### Message View

```
BMC SOFTWARE ----- IMS REGIONS ----- PERFORMANCE MGMT
SERV ==> REGNS          INPUT  10:38:57  INTVL=> 3  LOG=> N  TGT==> IMSA
PARAM ==> SO=NA, V=MSG          ROW    1 OF    4  SCROLL=> CSR
VIEW:    SM, MSG, DLI          EXPAND: FSEL(+), LINESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW

  NAME      TYP CL1 CL2 CL3 CL4  TRANCODE  PRLIM QUEUED M-DEQ M-GN M-PRG M-ISR
  -----
  CICS3302 DBT
  CICS3302 DBT          DLZZ          0      0      0      0      0      0
  CICS3302 DBT
  IMSM17X MPP    1    2    3    4
***** END OF DATA *****
```

### REGNS

#### DL/I View

```
BMC SOFTWARE ----- IMS REGIONS ----- PERFORMANCE MGMT
SERV ==> REGNS          INPUT  10:38:57  INTVL=> 3  LOG=> N  TGT==> IMSA
PARAM ==> SO=NA, V=DLI          ROW    1 OF    4  SCROLL=> CSR
VIEW:    SM, MSG, DLI          EXPAND: FSEL(+), LINESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW

  NAME      TRANCODE M-DEQ   GU   GN   GHU   GHN  REPL  ISRT  DLET   FP   TOT
  -----
  CICS3302
  CICS3302 DLZZ          0     0     0     0     0     0     0     0     0
  CICS3302
***** END OF DATA *****
```

#### Description:

This version of REGNS shows IMS dependent region activity and the work currently being performed by IMS for the duration of a PSB scheduling. The REGNS version shown in “REGNS - IMS Regions (Event Collector Data)” on page 164 shows region activity for the transaction currently processing. It is available only when the Event Collector is active.

REGNS arranges information by views. If the Event Collector is not active, only the following views of dependent region activity are available:

SM (summary)	Shows status information about your active regions and the transactions that are currently executing. This is the default display.
MSG (message)	Shows information related to scheduling and the message queue.
DLI	Shows information by region about DL/I database activity for a scheduling.

**Color:**

If you have a color monitor:

Blue	Indicates the current view.
Red	Highlights regions (rows) that are in a wait state.
Turquoise	Indicates normal values.
Yellow	Indicates an error message.
Pink	Indicates a WFI or pseudo-WFI transaction is waiting for input. This affects the STATUS, PSB, and TRANCODE columns.

**Select Code:**

REGNS

**Parameter:**

All the REGNS parameters, except for SORT and VIEW, act as filters that restrict the information shown according to the criteria specified by the parameter(s). SORT sorts the columns in the display by specified characters. VIEW shows the requested view of REGNS data. The REGNS parameters can be used as follows:

- The filtering parameters apply across all views. All views are restricted by the specified parameter.

For example, if you enter:

TYPE=DBT

in the DLI view, only the rows for DBCTL threads are displayed. A filtering parameter can be used with any view even though that view does not display the applicable field.

**Tuning Tip**

Use the NI or NOTIDLE parameter to see only those regions processing transactions.

- Multiple parameters must be separated by commas.
- A blank indicates the end of a parameter string.

- Multiple resources with similar names can be requested by using an \* character as a generic qualifier and a + character as a positional qualifier. The positional qualifier is repeated for every character to be replaced. The generic qualifier replaces a group of characters.

For example, a parameter of NAME=MP\* shows all region names that start with MP and have any character in the rest of the name.

- If multiple filtering parameters are entered, the regions displayed must meet all the restrictions.
- If one parameter invalidates another, an error message is issued without further processing.

The following parameter descriptions are arranged alphabetically. The descriptions begin with parameters that start with a special character. Parameters containing a numeric character are first in their alphabetical group.

AC|ACTCLASS=nnn|(nnn,...)

Where nnn represents a transaction class number. It displays in each view only those regions that are currently running a transaction in the specified class(es). A maximum of four classes can be specified.

AGN=xxxxxxxx

Where xxxxxxxxx represents the name of an application group. It displays in each view only those regions with a name that matches the specified application group name. The application group name for a region is shown in the AGN field of the DREGN display.

C1|CL1=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the first class equal to the specified number.

C2|CL2=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the second class equal to the specified number.

C3|CL3=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the third class equal to the specified number.

C4|CL4=nnn

Where nnn represents a transaction class number. It displays in each view only those regions with the fourth class equal to the specified number.

CL|CLASS=nnn|(nnn,...)

Where nnn represents a transaction class number. It displays in each view only those regions that can process this class(es) of transaction. A maximum of four classes can be specified.

DT|DLET<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a DL/I database delete count. It displays in each view only those regions with transactions that have a database delete count less than (<), greater than (>), or equal to (=) the specified number.

DL|DLI<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents the total DL/I call count. It displays in each view only those regions currently executing transactions with a total DL/I call count less than (<), greater than (>), or equal to (=) the specified number.

EL|ELAPSED<nnn|>nnn|=nnn

Where nnn represents elapsed time in seconds. It displays in each view only those regions currently executing transactions that have an elapsed time less than (<), greater than (>), or equal to (=) the specified number.

FP<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a Fast Path database call count. It displays in each view only those regions with transactions that have a Fast Path database call count less than (<), greater than (>), or equal to (=) the specified number.

GN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a GET NEXT and GET NEXT within PARENT count. It displays in each view only those regions with transactions that have a database GN and GNP count less than (<), greater than (>), or equal to (=) the specified number.

GU<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a GET UNIQUE count. It displays in each view only those regions with transactions that have a database GU count less than (<), greater than (>), or equal to (=) the specified number.

HN|GHN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a GET HOLD NEXT and GET HOLD NEXT within PARENT count. It displays in each view only those regions with transactions that have a database GHN and GHNP count less than (<), greater than (>), or equal to (=) the specified number.

HU|GHU<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a GET HOLD UNIQUE count. It displays in each view only those regions with transactions that have a database GHU count less than (<), greater than (>), or equal to (=) the specified number.

ID<nnn|>nnn|=nnn

Where nnn represents a region number. It displays in each view only those regions with a region number less than (<), greater than (>), or equal to (=) the specified number.

IS|ISRT<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents an INSERT count. It displays in each view only those regions with transactions that have a database ISRT count less than (<), greater than (>), or equal to (=) the specified number.

LO|LOCKS<nnnnnnnnnn|>nnnnnnnnnn| =nnnnnnnnnn

Where nnnnnnnnn represents a PI or IRLM enqueue count. It displays in each view only those regions currently executing transactions with either outstanding PI enqueues or IRLM locks (if IMS IRLM is active) less than (<), greater than (>), or equal to (=) the specified number.

LT=xxxxxxx

Where xxxxxxxx represents an LTERM name. It displays in each view only those regions processing transactions originating from an LTERM that matches the specified LTERM name. Multiple regions can be displayed by using an \* or + qualifier; for example, LT=SNL++060.

MD|M-DEQ<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message dequeue count. It displays in each view only those regions with transactions that have a message dequeue count less than (<), greater than (>), or equal to (=) the specified number.

MG|M-GN<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message GET NEXT count. It displays in each view only those regions with transactions with message GN counts less than (<), greater than (>), or equal to (=) the specified number.

MI|M-ISR<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message INSERT count. It displays in each view only those regions with transactions that have a message ISRT count less than (<), greater than (>), or equal to (=) the specified number.

MP|M-PRG<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a message PURGE count. It displays in each view only those regions with transactions that have a message PURG count less than (<), greater than (>), or equal to (=) the specified number.

NA|NAME=xxxxxxxx

Where xxxxxxxx represents a region name. It displays in each view only those regions with a name that matches the specified region name. Multiple regions can be displayed by using an \* or + qualifier; for example, NAME=\*RGN1.

NI|NOTIDLE

It displays in each view only those regions that are not idle (active).

PR|PRLIM<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a region processing limit count. It displays in each view only those regions with transactions that have a processing limit count less than (<), greater than (>), or equal to (=) the specified number.

PS|PSB=xxxxxxxx

Where xxxxxxxx represents a PSB name. It displays in each view only those regions with PSB names that match the specified name. Multiple regions can be displayed by using an \* or + qualifier; for example, PSB=SM++0+00.

QU|QUEUED<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a transaction message input queue count. It displays in each view only those regions with transactions that have a message input queue count that is less than (<), greater than (>), or equal to (=) the specified number.

RE|REPL<nnnnnnnn|>nnnnnnnn| =nnnnnnnn

Where nnnnnnnn represents a database REPLACE count. It displays in each view only those regions with transactions that have a database REPL count less than (<), greater than (>), or equal to (=) the specified number.

SO|SORT=cc

Where cc represents any of the following two-character view columns. The display is sorted by NAME by default. The sort sequence is ascending for alphanumeric characters and descending for numeric characters except for ID. The numeric region ID is sorted in ascending sequence.

**Tip**

An easy way to sort is to move the cursor to the column heading and press ENTER.

The following SORT parameter descriptions are arranged alphabetically. SORT keywords that start with a special character are described first. Keywords containing a numeric character are first in their alphabetical group. All views are sorted as requested by the specified parameter.

Any of these keywords can be used in any view.

- C1    Sorts the list by CL1 (descending).
- C2    Sorts the list by CL2 (descending).
- C3    Sorts the list by CL3 (descending).
- C4    Sorts the list by CL4 (descending).
- DA    Sorts the list by YES (DB2 thread is active) or NO (DB2 thread is not active). The order is descending with YES first in the sort.
- DL    Sorts the list by DLI (descending).
- DT    Sorts the list by DLET (DL/I database; descending).
- EL    Sorts the list by ELAPSED (DL/I database; descending).
- FP    Sorts the list by FP (DL/I database; descending).
- GN    Sorts the list by GN (DL/I database; descending).
- GU    Sorts the list by GU (DL/I database; descending).
- HN    Sorts the list by GHN (DL/I database; descending).
- HU    Sorts the list by GHU (DL/I database; descending).
- ID    Sorts the list by ID (numeric region ID; ascending).
- IS    Sorts the list by ISRT (DL/I database; descending).
- LO    Sorts the list by LOCKS (descending).
- LT    Sorts the list by USR/LTRM (ascending).
- MD    Sorts the list by M-DEQ (descending).
- MG    Sorts the list by M-GN (descending).
- MI    Sorts the list by M-ISRT (descending).
- MP    Sorts the list by M-PRG (descending).
- NA    Sorts the list by NAME (default; ascending).
- PL    Sorts the list by PRLIM (descending).
- PS    Sorts the list by PSB name (ascending).
- QU    Sorts the list by QUEUED (descending).
- RE    Sorts the list by REPL (DL/I database; descending).
- ST    Sorts the list by STATUS (ascending).
- TO    Sorts the list by TOT (DL/I database; descending).
- TR    Sorts the list by TRANCODE (ascending).
- TY    Sorts the list by TYP (ascending).
- US    Sorts the list by USR/LTRM (ascending).

ST|STATUS=

It displays in each view only those regions that meet the specified criteria. Multiple regions can be displayed by using an \* or + qualifier. For example, you can select all:

- Active regions by specifying ST=A\*
- Idle regions by specifying ST=I D\*
- Waiting regions by specifying ST=W\*

Any of the region statuses described on page 119 can be specified with the STATUS parameter.

TO|TOT<nnnnnnnn|>nnnnnnnn|=nnnnnnnn

Where nnnnnnnn represents a total DL/I database call count. It displays in each view only those regions with transactions that have a total DL/I database call count less than (<), greater than (>), or equal to (=) the specified number.

TR|TRANCODE=xxxxxxx

Where xxxxxxxx represents a transaction code. It displays in each view only those regions that have transaction codes that match the specified code. Multiple regions can be displayed by using an \* or + qualifier; for example, TR=+MZ200XX.

TY|TYP=xxx

Where xxx represents a region type. Multiple regions can be displayed by using an \* or + qualifier; for example, TYP=M\*.

The following type descriptions are arranged alphabetically:

BMI	Displays batch message processing regions currently executing an implicit APPC/IMS transaction.
BMO	Displays batch message processing regions currently executing an OTMA transaction.
BMP	Displays batch message processing regions.
BMW	Displays wait-for-input BMP.
DBT	Displays DBCTL CICS threads.
FPU	Displays Fast Path utility regions.
JBP	Displays Java batch message processing regions.
JMI	Displays Java message processing regions currently executing an implicit APPC/IMS transaction.
JMO	Displays Java message processing regions currently executing an OTMA transaction.
JMP	Displays Java message processing regions.
JMW	Displays wait-for-input JMP.
MDP	Displays message-driven Fast Path regions.
MPI	Displays message processing regions currently executing an implicit APPC/IMS transaction.
MPO	Displays message processing regions currently executing an OTMA transaction.

MPP	Displays message processing regions.
MPW	Displays wait-for-input MPP.
NDP	Displays non-message-driven Fast Path regions.
ODB	Displays DBCTL ODBA threads.
TPI	Displays message processing regions currently executing an explicit CPI-C program.

US|USR=xxxxxxx

Where xxxxxxxx represents a user ID. It displays in each view only those regions with user IDs that match the specified ID. Multiple regions can be displayed by using an \* or + qualifier; for example, USR=CI CS\*.

V|VIEW= SM|MSG|DLI

Displays the specified view:

SM	Displays the REGNS summary status view (default).
MSG	Displays the IMS/DC message view.
DLI	Displays the DL/I database view.

#### View Selection:

You can:

- Enter the VIEW parameter in the PARM field.
- Move the cursor to the view you want in the VIEW line and press ENTER.
- Use one of the following PF keys:

PF10          Displays the previous view.

PF11          Displays the next view.

#### Expand:

The REGNS display can be EXPANDED by using the following fields as indicated:

FSEL(+)

Indicates that at least one column has cursor-sensitive fields. A string of + characters underneath a column name means this column contains cursor-sensitive fields. Moving the cursor to a field in that column and pressing the ENTER key invokes a related display for that field. For example, if the cursor is at a field value under the DLI column, pressing the ENTER key displays the DLIST service with the correct region number as the parameter. Columns with cursor-sensitive fields include:

DLI

The DLIST service showing DL/I call details can be displayed with the region number as a parameter.

DB2      You can hyperlink from this field to MAINVIEW for DB2 as described on page 197.

#### LOCKS

If IMS IRLM is active, the LUSRD service showing IRLM lock user details can be displayed with the region number as a parameter. If IRLM is not active, the PI service showing program isolation information can be displayed with the region number as a parameter.

#### QUEUED

The CLASQ service showing transaction queuing and processing can be displayed with the region number as a parameter.

#### TOT

The DLIST service showing DL/I call details can be displayed with the region number as a parameter.

#### TRANCODE

The TRANQ service showing transaction status details can be displayed with the transaction code as a parameter.

You can also hyperlink from this field to MAINVIEW for CICS as described in the hyperlink section below.

When a display service is cursor-selected from a REGNS field, that service shows system data at the time it is invoked. It is not synchronized with the REGNS display.

Use the END PF key to return to the REGNS display.

#### LINESEL(DREGN)

Move the cursor to a region in one of the rows of a REGNS view and press ENTER to invoke the DREGN service. DREGN shows region details for the selected region.

Selecting DREGN with the cursor from the EXPAND line invokes the DREGN service for the region in the first row of the REGNS display.

#### MON(RGN)

Move the cursor to this field in the EXPAND line to display the Active Timer Request service for all active region monitors.

**Hyperlink:**

If MAINVIEW for CICS is installed and a transaction is a CICS transaction, you can link to the TASKXPND service in MAINVIEW for CICS by using the TRANCODE column in any view to cursor-select that transaction.

If MAINVIEW for DB2 is installed and a transaction has an active DB2 thread, you can link to DUSER in MAINVIEW for DB2 by using the DB2 column in the summary view to cursor-select that transaction.

**Sorting:**

The display list can be sorted by:

- Using the SORT parameter
- Positioning the cursor with the TAB key to the column heading to be sorted and pressing ENTER

This overrides any SORT parameter in the PARM field and primes that field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by NAME by default. Alphanumeric fields and the ID field are sorted in ascending order. Numeric fields are sorted in descending order. Fields with no values are sorted to the bottom of the list.

**Scrolling:**

The display is scrollable.

**Field Descriptions:**

Each of the fields is shown and described below by view.

Summary View											
BMC SOFTWARE -----				IMS REGIONS				-----PERFORMANCE MGMT			
SERV ==>	REGNS		INPUT	10:38:57	INTVL=>	3	LOG=>	N	TGT==>	IMSA	
PARM ==>	SO=NA, V=SM				ROW	1	OF	4	SCROLL=>	CSR	
VIEW:	SM, MSG, DLI				EXPAND:	FSEL(+),	LINESEL(DREGN),	MON(RGN)			
PF10/11 FOR PREV/NEXT VIEW											
ID	NAME	TYP	STATUS	PSB	TRANCODE	USR/LTRM	DLI	DB2	LOCKS	ELAPSED	
---	---	---	---	---	++++++	---	++++	---	---	---	---
1	CICS3302	DBT	IDLE								
2	CICS3302	DBT	ACTV-USR	STBCUSR	DLZZ		10		1		
3	CICS3302	DBT	IDLE	ABEM003T	ABMAIL	CIR11	4	YES			
***** END OF DATA *****											

This view shows the status of active regions and transactions. The descriptions are arranged in alphabetical order.

**DB2**

Indicates an active DB2 thread as:

YES            This transaction has an active DB2 thread.

NO            This transaction is not using DB2.

You can hyperlink from this column to the DUSER service in MAINVIEW for DB2 if it is installed and there is an active DB2 thread.

## DLI

Total DL/I database call count since PSB scheduling.

You can expand from this column to the DLIST service.

## ELAPSED

Current elapsed time calculated by subtracting transaction enqueue time from current time. If enqueue time is not available, this field is blank.

## ID

Region identifier assigned to this region by IMS.

## LOCKS

Either total outstanding PI enqueue counts or IRLM locks if IMS IRLM is active. An \* character following the count indicates that this transaction is waiting for a PI or IRLM lock if IMS IRLM is active.

You can expand from this column to the LUSRD service if IRLM is active. Otherwise, you can expand to the PI service.

## NAME

Job name or started task procedure name of the region.

## PSB

PSB name of the application program currently being processed.

## STATUS

Region status:

ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (the Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.

INACTIVE	<p>Region defined but not started (not yet signed on).</p> <p>The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.</p>
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	<p>Region waiting for an AO message.</p> <p>Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.</p>
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	<p>Region in scheduling (waiting for block mover latch).</p> <p>The block mover latch comprises several smaller latches. Use the LATCH service to determine which BML latch a region is waiting for.</p>
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.

WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

## TRANCODE

Name of transaction currently being processed.

For an IMS transaction, you can expand from this column to the TRANQ service.  
For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.

## TYP

Type of region that is processing the transaction:

BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.
BMO	Batch message processing region currently executing an OTMA transaction.
BMP	Batch message processing region.
BMW	Wait-for-input BMP.
DBT	DBCTL CICS thread.
FPU	Fast Path utility region.
JBP	Java batch message processing region.
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.
JMO	Java message processing region currently executing an OTMA transaction.
JMP	Java message processing region.
JMW	Wait-for-input JMP.
MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

## USR/LTRM

ID of the user who submitted this transaction or LTERM where transaction was submitted. An \* character indicates a terminal ID was used instead of a user ID.

```

Message View

BMC SOFTWARE -----          IMS REGIONS          -----PERFORMANCE MGMT
SERV ==> REGNS                INPUT  10:38:57  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==> SO=NA, V=MSG                                ROW    1 OF    4 SCROLL=> CSR
VIEW: SM, MSG, DLI                                EXPAND: FSEL(+), LI NESEL(DREGN), MON(RGN)
PF10/11 FOR PREV/NEXT VIEW

NAME      TYP  CL1  CL2  CL3  CL4  TRANCODE  PRLIM  QUEUED  M-DEQ  M-GN  M-PRG  M-ISR
-----
CICS3302 DBT
CICS3302 DBT          DLZZ          0          0          0          0          0
CICS3302 DBT
IMSM17X  MPP      1    2    3    4

***** END OF DATA *****

```

This view shows transaction activity in each active IMS dependent region. The descriptions are arranged in alphabetical order.

For the following fields, CL1 through CL4, the currently active class is shown by a > character preceding the class:

### CL1

First class this MPP can process. (This field does not apply for a TYP of DBT and for all Fast Path types, including MDP and NDP.)

### CL2

Second class this MPP can process. (This field does not apply for a TYP of DBT and for all Fast Path types, including MDP and NDP.)

### CL3

Third class this MPP can process. (This field does not apply for a TYP of DBT and for all Fast Path types, including MDP and NDP.)

### CL4

Fourth class this MPP can process. (This field does not apply for a TYP of DBT and for all Fast Path types, including MDP and NDP.)

### M-DEQ

Number of messages successfully processed by the application program in this scheduling. For MDPs (message-driven program), this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

### M-GN

Number of MESSAGE GET NEXT calls issued by this program since PSB scheduling. For MDPs, this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

M-ISR	Number of message insert calls issued by this program since PSB scheduling. For MDPs, this value does not include I/O PCB ISRTs. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
M-PRG	Number of message purge calls issued by this program since PSB scheduling. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
NAME	Job name or started task procedure name of the region.	
PRLIM	Processing limit count of the transaction. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
QUEUED	Number of currently queued input messages with this transaction code. For MDPs (message-driven program), the number of messages queued on this region's BALG. You can expand from this column to the CLASQ service. (This field does not apply to DBCTL threads, region types DBT and ODB.)	
TRANCODE	Name of transaction currently being processed. For an IMS transaction, you can expand from this column to the TRANQ service. For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.	
TYP	Type of region processing a transaction:	
BMI	Batch message processing region currently executing an implicit APPC/IMS transaction.	
BMO	Batch message processing region currently executing an OTMA transaction.	
BMP	Batch message processing region.	
BMW	Wait-for-input BMP.	
DBT	DBCTL CICS thread.	
FPU	Fast Path utility region.	
JBP	Java batch message processing region.	
JMI	Java message processing region currently executing an implicit APPC/IMS transaction.	
JMO	Java message processing region currently executing an OTMA transaction.	
JMP	Java message processing region.	
JMW	Wait-for-input JMP.	

MDP	Message-driven Fast Path region.
MPI	Message processing region currently executing an implicit APPC/IMS transaction.
MPO	Message processing region currently executing an OTMA transaction.
MPP	Message processing region.
MPW	Wait-for-input MPP.
NDP	Non-message-driven Fast Path region.
ODB	DBCTL ODBA thread.
TPI	Message processing region currently executing an explicit CPI-C program.

<u>DL/I View</u>											
BMC SOFTWARE -----				IMS REGIONS				-----PERFORMANCE MGMT			
SERV ==>	REGNS	INPUT	10:38:57	INTVL=>	3	LOG=>	N	TGT==>	IMSA		
PARM ==>	S0=NA, V=DLI					ROW	1 OF	4	SCROLL=>	CSR	
VIEW:	SM, MSG, DLI			EXPAND:	FSEL(+),	LINESEL(DREGN),	MON(RGN)				
PF10/11 FOR PREV/NEXT VIEW											
NAME	TRANCODE	M-DEQ	GU	GN	GHU	GHN	REPL	ISRT	DLET	FP	TOT
-----	++++++	-----	-----	-----	-----	-----	-----	-----	-----	-----	++++
CICS3302											
CICS3302	DLZZ	0	0	0	0	0	0	0	0	0	0
CICS3302											
***** END OF DATA *****											

This view shows transaction DL/I database activity in each active IMS dependent region. The descriptions are arranged in alphabetical order.

#### M-DEQ

Number of messages successfully processed by the application program since PSB scheduling. For MDPs (message-driven program), this value is 0. (This field does not apply to DBCTL threads, region types DBT and ODB.)

#### NAME

Job name or started task procedure name of the region.

#### TRANCODE

Name of transaction currently being processed.

For an IMS transaction, you can expand from this column to the TRANQ service. For a CICS transaction, you can hyperlink to the TASKXPND service in MAINVIEW for CICS if it is installed.

For the following fields, the counts do not include calls to DEDBs and MSDBs. Fast Path calls are totaled in the FP column.

#### DLET

Total delete calls since PSB scheduling.

FP	Total calls to Fast Path databases since PSB scheduling.
GHN	Sum of GET HOLD NEXT and GET HOLD NEXT within PARENT calls since PSB scheduling.
GHU	Total GET HOLD UNIQUE calls since PSB scheduling.
GN	Sum of GET NEXT and GET NEXT within PARENT calls since PSB scheduling.
GU	Total database GET UNIQUE calls since PSB scheduling.
ISRT	Total insert calls since PSB scheduling.
REPL	Total replace calls since PSB scheduling.
TOT	Total DL/I database calls since PSB scheduling.
	From this column, you can expand to the DLIST service.

## STAT/STATR - System Status

BMC SOFTWARE----- SYSTEM STATUS ----- PERFORMANCE MGMT									
SERV ==>	STATR	INPUT	13:21:09	INTVL=>	3	LOG=>	N	TGT==>	IMSxxx
PARM ==>	RA0600I - ENTER FIRST REGION NUMBER					SCROLL => N/A			
RGN	1	2	3						
TYPE	DBT	DBT	BMP						
STAT	IDLE	IDLE	ACTV						
DB2	CON								
WARN									
--IMS DATA SETS--	TOTAL FREE		%	ALLOCATED		%	*****WARNING****		
SHORT	8,675	8,652	100	23	0	(HI =	18)		
LONG		1,025		972	95	53	5	(HI =	57)
QBLKS		19,700		19,653	100	47	0	(HI =	4)
ACBLIBA	MODBLKSB	SYSTEM: DB/DC		ETO: ACTIVE		DLQT: 5		SUSPEND QUEUE	
FORMATA	DFSOLP00	RELEASE: 6.1		APPC: ENABLED		-IMSLU62			
--TRACES--	DL/I..	LOCK..	DLOG..	DISP..	SCHD..	SUBS..	LATC...	PI....	FP... MON.. TIMEOUT
	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF AUTO 5
--STOPPED RESOURCES--	LINES:		6	TRANS:	0	DB'S:	33	RTCDE'S: 0	
	NODES:		0	PGMS:	0	APPCLU:	0		
--RESTART DATA--	12/15/02		10:29:15	DFSVNUC9	--LOCK MANAGER-- IRLM IR15				
--SCHEDULING--	FAILURES:		0	0%					
CLASS	001	002	003	004	005	006	REST	TOTAL	FP
Q'D	0	0	0	0	0	0	0	0	0
PROC	245	143	0	0	0	0	0	388	0

**Description:**

This display provides an overview of the status of the total IMS system (the work performed and consumption of resources) with indicators and highlighting of possible problem areas.

**Note:** STAT and STATR are equivalent except for the STOPPED RESOURCES lines that are shown only with STATR. STAT executes faster than STATR on large IMS systems with many resources defined.

**Select Code:**

STAT or STATR

**Parameter:**

No entry or enter 1 to display regions 1 through 15.

Enter n to display regions n through n+14, where n is a valid region identification number within the range of defined regions.

Enter 0 to display active regions only.

**Field Descriptions:**

Each of the fields is shown and described below by display area.

Area 1			
RGN	1	2	3
TYPE	MPP	MPP	BMP
STAT	IDLE	IDLE	ACTV
DB2	CON		
WARN			

The first area presents a concise summary of the status of 15 dependent regions. If any current or potential problem is detected, it is indicated in the corresponding region column on a warning line (for example, region in a program isolation wait, dynamic backout would fail, and so on).

The descriptions are arranged in alphabetical order.

DB2	DB2 Connection status:	
	blank	No connection to DB2
	CON	Region is connected to DB2
	SON	Region has signed on to DB2 and established a connection
	THD	Thread (unit of work) is active in DB2
RGN	Region number	
STAT	Region status:	
	blank	Undefined
	ACTV	Active in the application program
	BACK	Dynamic backout in progress
	BG-W	Fast Path MDP (message-driven program) waiting for input on BALG
	DB2	Active in DB2; a dependent region is executing an SQL call
	DL/I	Active in DL/I
	I-HT	Pseudo WFI region waiting for recurrence of the same transaction code. See the IBM publication, <i>IMS System Definition Reference</i> , for more information.
	IDLE	Started but not processing
	INAC	Inactive (defined but not started)
	OPEN	In open processing
	SCHD	Scheduling: waiting for some resource
	TERM	In region termination
	WF-W	WFI region waiting for input
TYPE	Region type:	
	blank	Undefined or inactive
	BMP	Batch message processing
	BWFI	BMP wait-for-input
	DBT	DBCTL CICS thread
	FPU	Fast Path utility
	JBP	Java batch message processing
	JMP	Java message processing
	JWFI	JMP wait-for-input
	MDP	Message-driven Fast Path
	MPP	Message processing
	MWFI	MPP wait-for-input
	NDP	Non-message-driven Fast Path
	ODB	DBCTL ODBA thread
	TPI	MPP region currently executing a CPI-C driven program

WARN	Warning. Possible problem indicators:	
BACK	Dynamic backout is in progress.	
BKNG	Dynamic backout failed; IMS will ABEND.	
NOBK	Dynamic backout is no longer valid for this region. The dynamic log has wrapped and this region was unable to log all database changes. If an ABEND, pseudo-ABEND, or roll call occurs, IMS ABENDs.	
W-FB	Region is waiting for a Fast Path fixed buffer.	
W-FC	Region is waiting for a Fast Path control interval (CI contention).	
W-FO	Region is waiting for the Fast Path overflow buffer latch (OBA latch).	
W-FR	Region is waiting for the Fast Path resource latch.	
W-FS	Region is waiting for the Fast Path sync-point latch.	
W-IR	Region is in IRLM wait (IRLM must be active) for a resource, belonging to this or another IMS, that is being held by another region.	
W-LT	Region is waiting for a CLLE latch	
W-NF	Region is waiting for asynchronous notify(s) to complete.	
W-PI	Region is in a program isolation wait for a resource held by another region.	
W-ST	Region is waiting for storage in one of the IMS pools.	

<u>Area 2</u>							
--IMS DATA SETS--	TOTAL	FREE	%	ALLOCATED	%	****WARNING****	
SHORT	8,675	8,652	100	23	0	(HI=	18)
LONG	1,025	972	95	53	5	(HI=	57)
QBLKS	19,700	19,653	100	47	0	(HI=	4)

The second area shows the current status of the queue data sets. A warning is indicated on the right if a problem is detected (for example, a data set usage threshold is reached).

The descriptions are arranged in alphabetical order by row.

#### **--IMS DATA SETS--**

##### **LONG**

Long message queue data set utilization. (This field does not apply to DBCTL threads, region types DBT and ODB.)

##### **ALLOCATED %**

Number of records that are allocated and the percentage of the total records allocated.

##### **FREE %**

Number of records that are free and the percentage of the total records free.

##### **TOTAL**

Total number of records.

\*\*\*\*WARNING\*\*\*\*

Indicates whether the QTU threshold has been reached as follows:

When the % value for ALLOCATED records for the long message queue data set exceeds the threshold specified by the QTU parameter for the IMS control region, THRESHOLD is displayed and highlighted in this field.

If the % value for ALLOCATED records for the long message queue data set does not exceed the threshold specified by the QTU parameter, the high-water mark is shown in this field.

If CHKPT PURGE, CHKPT DUMPQ, or CHKPT FREEZ is displayed, SCHEDULING STOPPED is shown in Area 5 of the display as described on page 214.

**Note:** (IMS DB/DC only)

The ALLOCATED total may exceed the high-water mark early in an IMS session. This happens because the ALLOCATED totals include bit-map and message-queue records that are rebuilt into the current IMS at restart. As these records are put into actual use, the high-water marks (highest record number) eventually adjust.

QBLKS

QBLKS message queue data set utilization. (This field does not apply to DBCTL threads, region types DBT and ODB.)

ALLOCATED %

Number of records that are allocated and the percentage of the total records allocated.

FREE %

Number of records that are free and the percentage of the total records free.

TOTAL

Total number of records.

\*\*\*\*WARNING\*\*\*\*

Indicates whether the QTU threshold has been reached as follows:

When the % value for ALLOCATED records for the QBLKS message queue data set exceeds the threshold specified by the QTU parameter for the IMS control region, THRESHOLD is displayed and highlighted in this field.

If the % value for ALLOCATED records for the long message queue data set does not exceed the threshold specified by the QTU parameter, the high-water mark is shown in this field.

If CHKPT PURGE, CHKPT DUMPQ, or CHKPT FREEZ is displayed, SCHEDULING STOPPED is shown in Area 5 of the display as described on page 214.

**Note:** IMS DB/DC only.

The ALLOCATED total may exceed the high-water mark early in an IMS session. This happens because the ALLOCATED totals include bit-map and message-queue records that are rebuilt into the current IMS at restart. As these records are put into actual use, the high-water marks (highest record number) eventually adjust.

## SHORT

Short message queue data set utilization. (This field does not apply to DBCTL threads, region types DBT and ODB.)

### ALLOCATED %

Number of records that are allocated and the percentage of the total records allocated.

### FREE %

Number of records that are free and the percentage of the total records free.

### TOTAL

Total number of records.

### \*\*\*\*WARNING\*\*\*\*

Indicates whether the QTU threshold has been reached as follows:

When the % value for ALLOCATED records for the short message queue data set exceeds the threshold specified by the QTU parameter for the IMS control region, THRESHOLD is displayed and highlighted in this field.

If the % value for ALLOCATED records for the long message queue data set does not exceed the threshold specified by the QTU parameter, the high-water mark is shown in this field.

If CHKPT PURGE, CHKPT DUMPQ, or CHKPT FREEZ is displayed, SCHEDULING STOPPED is shown in Area 5 of the display as described on page 214.

**Note:** IMS DB/DC only.

The ALLOCATED total may exceed the high-water mark early in an IMS session. This happens because the ALLOCATED totals include bit-map and message-queue records that are rebuilt into the current IMS at restart. As these records are put into actual use, the high-water marks (highest record number) eventually adjust.

### Area 3

ACBLIBA MODBLKSB      SYSTEM: DB/DC      ETO: ACTIVE      DLQT: 5 SUSPEND QUEUE

FORMATA DFSOLP00      RELEASE 6.1      APPC: ENABLED      1 RELEASE

The third area shows the ddnames in use, IMS system type, current release, status of the ETO feature, dead letter queue time in days, and the status of the APPC feature.

The descriptions are arranged in alphabetical order.

#### ACBLIBA MODBLKSA

Current IMS ddnames in use.

#### APPC

This field is highlighted if the status is STOPPED or different from the requested status.

The value can be either INACTIVE or in the form of xxxxxxxx -nnnnnnnn, where:

xxxxxxx      Is the IMS status, which can be one of the following:

- CANCELED
- DISABLED
- ENABLED
- FAILED
- OUTBOUND
- PURGING
- STARTING
- STOPPED

See IBM's *IMS Operator Reference* for a description of the status in the “/DIS APPC” section.

nnnnnnnn      Is the IMS base LU6.2 name (if known)

#### DLQT

Shows the Dead Letter Queue Time (DLQT) in days for this IMS. This value is specified at IMS startup and represents the number of days that an ETO user structure can go unreferenced before it is considered to be in a dead queue status. (The field is blank for DBCTL targets.)

#### ETO

This field shows the status of the ETO feature as ACTIVE or INACTIVE. (The field is blank for DBCTL targets.)

#### FORMATA DFSOLP00

Current IMS ddnames in use.

#### RELEASE

IMS system release.

#### SUSPEND QUEUE

This value is displayed and highlighted in the WARNING column if there are any transactions on the suspend queue. It is not displayed if there are no transactions on the suspend queue.

## SYSTEM

IMS system type:

DB/DC  
DBCTL  
DCCTL

### Area 4

```
-- TRACES--  DL/I . . LOCK . . DLOG . . DISP . . SCHD . . SUBS . . LATC . . . PI . . . FP . . MON . . TIMEOUT
              ON   ON   ON   ON   ON   OFF   OFF   OFF   OFF   OFF   AUTO  5
-- STOPPED RESOURCES--  LINES:   6  TRANS:   0  DB'S:   33  RTCDE'S:   0
                        NODES:    0   PGMS:    0  APPCLU:    0
```

The fourth area shows the current status of the internal traces, the total number of stopped lines, transactions, databases, routing codes, VTAM nodes, and programs (PSBs).

**Note:** STOPPED RESOURCES is available only with the STATR service. STAT shows blanks in this area.

The descriptions are arranged in alphabetical order.

### --TRACES--

This section displays the status of IMS traces.

The trace abbreviations for internal traces are the same as those used in the /TRA IMS command. The trace status for internal traces is shown as:

OFF            IMS internal trace is not active.  
ON             IMS internal trace is active.

The VTAM timeout trace has special status indicators as follows:

TIMEOUT        VTAM timeout trace. The format is:

      SSSS mm

      where:

      SSSS            Is the status of the VTAM I/O timeout trace. Possible values:

AUTO	Timeout trace active with AUTO option
MSG	Timeout trace active with MSG option
OFF	Timeout trace not active
ON	Timeout trace active; neither MSG or AUTO option specified

      mm             Is the timeout threshold in minutes, 0 (no timeout limit), or blank (trace is not active).

## **--STOPPED RESOURCES--**

The values shown in this area are determined according to the rules of the /DISPLAY STATUS command; they are available only with the STATR service. The lines are blank in the STAT service.

The descriptions are arranged in alphabetical order.

APPCLU	The number of APPC LUs that are stopped or have an associated, stopped TP (transaction program). This field is blank for DBCTL targets.
DB'S	The number of databases currently stopped (stopped, locked, not open, inquiry only). Uninitialized databases are not included in this count.
LINES	The number of communication lines currently stopped (no input, no output, no queuing, idle). (This field does not apply to DBCTL threads, region types DBT and ODB.)
NODES	VTAM stopped nodes. (This field does not apply to DBCTL threads, region types DBT and ODB.)
PGMS	Number of programs (PSBs) currently stopped.
RTCDE'S	The number of routing codes currently stopped if Fast Path is installed. (This field does not apply to DBCTL threads, region types DBT and ODB.)
TRANS	The number of transaction codes currently stopped (no queuing, no scheduling, locked, locked for /DBD).

### **Area 5**

--RESTART DATA--	12/15/02	11:28:52	DFSVNUC8	--LOCK MANAGER--	IRLM IR15				
--SCHEDULING--	FAILURES:	0	0%						
CLASS	001	002	003	004	005	006	REST	TOTAL	FP
Q'D	0	0	0	0	0	0	0	0	0
PROC	245	143	0	0	0	0	0	388	0

The fifth area shows scheduling, the number of failures, the current status, how many transactions are currently queued, and how many have been processed.

The descriptions are arranged in alphabetical order.

## **--RESTART DATA--**

**Note:** When IMS 6.1 or later is running, UTC (Universal Time Clock) time is used.  
When IMS 5.1 is running, local time is used.

12/15/02

The restart date (mm/dd/yy).

11:28:52

The IMS restart time (hh:mm:ss).

## DFSVNUC2

Current IMS nucleus loaded.

## LOCK MANAGER

Displays the current IMS lock manager. Possible values:

IRLM ssi d	Locks are managed by IRLM with subsystem ID ssi d.
PI	Locks are managed by Program Isolation.

## --SCHEDULING--

### CLASS

Transaction class. The counts for classes 001 through 006 are shown separately; higher classes are summarized under REST. Totals are also shown. If Fast Path is installed, the counts are summarized under FP. (This field does not apply to DBCTL threads, region types DBT and ODB.)

### FAILURES

The number of unsuccessful attempts at scheduling and the percent this represents of total activity.

**Note:** Because the counts of the different types of failures are kept in halfword counters, this sum is invalid if any one of these exceeds 65,535.

### PROC

The number of transactions processed since the last IMS cold start.

**Note:** These counts are accumulated from halfword counters kept by transaction code. If more than 32,767 transactions of one type are processed, this counter wraps to zero and the accumulation is lower than the true total.

### Q'D

The number of transactions currently queued. An asterisk appears to the right of the count if any enqueued transactions within that class are not schedulable (locked, stopped, or priority zero). (This field does not apply to DBCTL threads, region types DBT and ODB.)

### SCHEDULING STOPPED

This value is displayed to the right of FAILURES if scheduling is not currently active. The reason is shown in the WARNING column (see the Area 2 description):

CHKPT DUMPQ	A checkpoint DUMPQ is in progress.
CHKPT FREEZE	A checkpoint FREEZE is in progress.
CHKPT PURGE	A checkpoint PURGE is in progress.

# Chapter 13. Database Displays

This chapter describes the displays that show the status, activity, and performance of IMS resources as programs execute in the dependent regions, accessing message queues and various databases (DL/I or DB2).

## DBST - OSAM Global Pool Statistics

BMC SOFTWARE	-----	OSAM POOLS	-----	PERFORMANCE MGMT
SERV ==> DBST		INPUT 13:21:09	INTVL=> 3	LOG=> N
PARM ==>	RA1307I	ENTER SUBPOOL TO BE DISPLAYED		TGT==> IMSxxx
				SCROLL => N/A
-----				
OSAM DATABASE POOL STATISTICS				
-----				
460,800	TOTAL POOL SIZE	8	SUBPOOLS	DL/I TRACE IS ON
-----				
OSAM BUFFER HANDLER REQUESTS				
-----				
1,946	SEARCHES	140	ALTERS	
198	RETRIEVES BY KEY	0	BLOCK CREATES	
0	GET NEXTS	70	PURGES	
-----				
OSAM BUFFER HANDLER PERFORMANCE				
-----				
117	OSAM READS WITH I/O	0	OSAM WRITES - STEAL	
1,809	SATISFIED FROM POOL	28	OSAM WRITES - PURGE	
93.9	HIT RATIO	98	BLOCKS WRITTEN - TOTAL	
0	ERROR BUFFERS IN POOL	0	FORMAT CYLINDER	
		0	BLOCKS CREATED	
		0	MAX ERROR BUFFERS IN POOL	
-----				
SEQUENTIAL BUFFERING STATISTICS				
-----				
	STATUS: ACTIVE	0K	STORAGE CURRENTLY IN USE	
100K	STORAGE LIMIT	66K	MAX STORAGE USED	

**Description:** This display shows global statistics, accumulated since IMS restart, for the whole pool.

**Select Code:** DBST

Global pool statistics.

**Parameter:** None for this global display.

Entry of a valid subpool number returns the subpool display (see “DBST - OSAM Subpool Statistics” on page 219).

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>				
-----				
OSAM DATABASE POOL STATISTICS				
-----				
460,800	TOTAL POOL SIZE	8	SUBPOOLS	DL/I TRACE IS ON

This area shows the current pool definition. The descriptions are arranged in alphabetical order.

DL/I TRACE IS ON/OFF

The status of the DL/I buffer handler trace is shown as either ON or OFF.

SUBPOOLS

The number of subpools.

TOTAL POOL SIZE

The total pool size in bytes (including all overhead).

**Area 2**

```
----- OSAM BUFFER HANDLER REQUESTS -----
1,946 SEARCHES                                140 ALTERS
198 RETRIEVES BY KEY                          0 BLOCK CREATES
0 GET NEXTS                                    70 PURGES
```

This area displays the total requests made to the buffer handler. The descriptions are arranged in alphabetical order.

ALTERS

The number of mark buffer altered requests.

BLOCK CREATES

These requests are made when it is necessary to format a new block in the data set.

GET NEXTS

The number of get nexts (incremented only for databases accessed through OSAM).

PURGES

The number of purge requests required for synchronization point or checkpoint processing.

RETRIEVES BY KEY

The number of retrieves by key (incremented only for databases accessed through OSAM).

SEARCHES

Includes block locates, byte locates, and searches to get or free space.

**Area 3**

```
----- OSAM BUFFER HANDLER PERFORMANCE -----
117 OSAM READS WITH I/O                      0 OSAM WRITES - STEAL
1,809 SATISFIED FROM POOL                     28 OSAM WRITES - PURGE
93.9 HIT RATIO                                98 BLOCKS WRITTEN - TOTAL
0 ERROR BUFFERS IN POOL                      0 FORMAT CYLINDER
                                              0 BLOCKS CREATED
                                              0 MAX ERROR BUFFERS IN POOL
```

This area displays the actual work performed by the buffer handler to satisfy the above requests. The descriptions are arranged in alphabetical order.

#### BLOCKS CREATED

The number of new blocks added to the data set.

#### BLOCKS WRITTEN - TOTAL

Total blocks written, including OSAM WRITES - STEAL and OSAM WRITES - PURGE.

#### ERROR BUFFERS IN POOL

The number of error buffers currently in the pool.

#### FORMAT CYLINDER

The number of new cylinders formatted in the data set.

#### HIT RATIO

The hit ratio for all OSAM buffer pools combined. It is a measure of buffer pool performance. A high hit ratio means that a high percentage of reads were satisfied from the buffer pool without having to access external storage.

Hit ratio is computed as follows:

$$100 \times (A / (A + B))$$

where:

A            Requests satisfied from pool

B            OSAM reads with I/O

#### MAX ERROR BUFFERS IN POOL

The largest number of error buffers ever in the pool since IMS restart.

#### OSAM READS WITH I/O

The number of reads performed by OSAM in the overflow area to retrieve the next sequential block. Each read counted here indicates that the buffer requested was not in the OSAM buffer pool and had to be read from external storage.

#### OSAM WRITES - PURGE

This counter is incremented when the pool is purged because of synchronization point or checkpoint processing. Each write may cause multiple blocks to be written.

#### OSAM WRITES - STEAL

The counter is incremented whenever a buffer is written to make space in the pool. These writes are very expensive and should be avoided. They indicate that your buffer pools may be too small.

#### SATISFIED FROM POOL

This counter is incremented when the request is satisfied with a record already in the pool.

<b>Area 4</b>	
----- SEQUENTIAL BUFFERING STATISTICS -----	
STATUS: ACTIVE	0K STORAGE CURRENTLY IN USE
100K STORAGE LIMIT	66K MAX STORAGE USED

This area displays sequential buffering statistics. It is blank for IMS releases prior to IMS 2.2. The descriptions are arranged in alphabetical order.

- MAX STORAGE USED**  
 High-water mark for storage used (in K) by sequential buffering for this run of IMS.
- STATUS**  
 Status of sequential buffering in this IMS system. Possible values:
  - ACTIVE**  
 Sequential buffering is enabled.
  - NOT INITIALIZED**  
 Sequential buffering option was not selected.
  - SOFTWARE PROBLEM**  
 Error with sequential buffering component.
  - STOPPED**  
 Sequential buffering was active but has been stopped with an operator command.
- STORAGE CURRENTLY IN USE**  
 Number of kilobytes of storage currently used by sequential buffering for all regions.
- STORAGE LIMIT**  
 Maximum amount of storage (in K) that can be allocated by sequential buffering. If no limit was specified, NO STORAGE LIMIT is displayed.

# DBST - OSAM Subpool Statistics

BMC SOFTWARE		----- OSAM POOLS -----				PERFORMANCE MGMT			
SERV ==> DBST		INPUT 13:21:09		INTVL=> 3		LOG=> N		TGT==> IMSxxx	
PARM ==> 7		RA1307I ENTER SUBPOOL TO BE DISPLAYED				SCROLL=> N/A			
----- OSAM SUBPOOL STATISTICS -----									
ID CP1	BUFFER SIZE	8,192	7 BUFFERS	PAGEFIX: BUFFERS N PREFIXES N					
----- OSAM SUBPOOL REQUESTS -----									
879 LOCATES				52 ALTERS					
65 PURGES				0 BLOCK CREATES					
----- OSAM SUBPOOL PERFORMANCE -----									
103 OSAM READS WITH I/O				0 OSAM WRITES - STEAL					
774 SATISFIED FROM POOL				26 BLOCKS WRITTEN - PURGE					
88.3 HIT RATIO				1,058 BUFFERS SEARCHED					
0 LOCATE CALLS WHICH WAITED FOR BUSY IDENTs									
0 BUFFER STEAL WAITS FOR BUFFER BUSY WRITING									
0 BUFFER STEAL WAITS FOR BUFFER BUSY READING									
0 BUFFER STEAL OR PURGE WAITS FOR OWNERSHIP TO BE RELEASED									
0 BUFFER STEAL WAITS BECAUSE NO BUFFER WAS AVAILABLE TO BE STOLEN									
LEVEL	0	1	2	3	4	5	6	7	8
BUFFERS STOLEN	92	12	0	0	0	0	0	0	0

- Description:

Statistics on OSAM buffer pool management are also kept by subpool, not just for the whole pool. These statistics are cumulative from IMS restart.
- Select Code:

DBST
- Parameter:

Enter subpool number (1 or 2 digits).
- If no entry is made, the global pool display (see “DBST - OSAM Global Pool Statistics” on page 215) is returned.
- Field Descriptions:

Each of the fields is shown and described below by display area.

<u>Area 1</u>									
----- OSAM SUBPOOL STATISTICS -----									
ID CP1	BUFFER SIZE	8,192	7 BUFFERS	PAGEFIX: BUFFERS N PREFIXES N					

- This area shows the current subpool definition. The descriptions are arranged in alphabetical order.
- BUFFERS

Number of buffers specified.
- BUFFER SIZE

The subpool buffer size in bytes.

## ID

Four-character subpool identification from the ID field on the IOBF control statement. Used when defining multiple subpools of the same size.

N/A if not used.

## PAGEFIX: BUFFERS

Indicates (Y/N) whether the long term page fixing option was chosen for data buffers in this pool.

## PAGEFIX: PREFIXES

Indicates (Y/N) whether the long term page fixing option was chosen for buffer prefixes in this pool.

<b><u>Area 2</u></b>	
----- OSAM SUBPOOL REQUESTS -----	
879 LOCATES	52 ALTERS
65 PURGES	0 BLOCK CREATES

This area displays the number of subpool requests. However, at this time, only the number of purge requests is being incremented by IMS. The descriptions are arranged in alphabetical order.

## ALTERS

The number of alters.

## BLOCK CREATES

The number of blocks created.

## LOCATES

The number of locates.

## PURGES

The number of purges.

<b><u>Area 3</u></b>	
----- OSAM SUBPOOL PERFORMANCE -----	
103 OSAM READS WITH I/O	0 OSAM WRITES - STEAL
774 SATISFIED FROM POOL	26 BLOCKS WRITTEN - PURGE
88.3 HIT RATIO	1,058 BUFFERS SEARCHED

This area displays the actual work performed by the buffer handler in this subpool. The descriptions are arranged in alphabetical order.

## BLOCKS WRITTEN - PURGE

This counter is incremented by the number of blocks purged from the pool in synchronization point or checkpoint processing. The number of OSAM WRITES/STEAL may be added to this to get the total number of blocks written.

#### BUFFERS SEARCHED

The total number of buffers searched to find the records. Both this counter and SATISFIED FROM POOL exclude work done for keyed requests; therefore, the sum of the subpool counters is less than the global counter displayed for the whole pool.

#### HIT RATIO

This value is a measure of buffer pool performance. A high hit ratio means that a high percentage of reads were satisfied from the buffer pool without having to access external storage.

Hit ratio is computed as follows:

$$100 \times (A / (A + B))$$

where:

A            Requests satisfied from pool

B            OSAM reads with I/O

#### OSAM READS WITH I/O

The number of OSAM reads. Each read counted here indicates that the buffer requested was not in the OSAM buffer pool and had to be read from external storage.

#### OSAM WRITES - STEAL

This counter is incremented whenever a buffer is written to make space in the pool. These writes are very expensive and should be avoided. They indicate that your buffer pool may be too small.

#### SATISFIED FROM POOL

This counter is incremented when the request is satisfied with a record already in the subpool.

#### Area 4

O LOCATE CALLS WHICH WAITED FOR BUSY IDENTs  
O BUFFER STEAL WAITS FOR BUFFER BUSY WRITING  
O BUFFER STEAL WAITS FOR BUFFER BUSY READING  
O BUFFER STEAL OR PURGE WAITS FOR OWNERSHIP TO BE RELEASED  
O BUFFER STEAL WAITS BECAUSE NO BUFFER WAS AVAILABLE TO BE STOLEN

This area displays five counters of waits that may occur in the buffer handler. The reason for each type of wait is documented in the display.

#### **Area 5**

LEVEL	0	1	2	3	4	5	6	7	8
BUFFERS STOLEN	92	12	0	0	0	0	0	0	0

This area presents an overview of the work performed to free space in the buffer pool. The descriptions are arranged in alphabetical order.

#### **BUFFERS STOLEN**

The number of buffers taken at each level.

#### **LEVEL**

The nine levels represent a difference in the amount of work necessary to free the buffer.

- A buffer at level 0 or 1 can be used immediately because it is unowned and unaltered.
- A buffer at level 2 or 3 can be used when a write that has already been initiated is complete.
- At level 4 or 5, the buffer must first be written out.
- Levels 0, 2, 4 indicate that the last owner of the buffer is the PST currently requesting one, while buffers at levels 1, 3, 5 belonged to another PST.
- Ownership must be released before a buffer at levels 6, 7, or 8 may be stolen.
- At level 6 the buffer is not marked altered; at level 7 it has been altered; and at level 8 the data is currently being read in.

# FPBST - Fast Path Buffer Pool Statistics

BMC SOFTWARE	-----	SERVICE FAST PATH BUFFER POOL	-----	PERFORMANCE MGMT
SERV ==> FPBST		INPUT 13: 21: 09	INTVL=> 3	LOG=> N
PARM ==>				TGT==> IMSxxx
				SCROLL => N/A
FAST PATH BUFFER POOL STATISTICS				
30 TOTAL BUFFERS (DBBF)				25 FIXED BUFFERS
10 TOTAL PRE-FIXED BUFFERS (DBFX)				0 BUFFERS IN USE BY PROGRAMS
2, 048 BUFFER SIZE (BSIZ)				1 BUFFERS BEING WRITTEN
				24 FIXED BUFFERS AVAILABLE
				5 UNFIXED BUFFERS
2 TOTAL OUTPUT THREADS (OTHR)				2 IDLE OUTPUT THREADS
61, 440 TOTAL FAST PATH BUFFER POOL SIZE				
51, 200 TOTAL PAGE-FIXED BUFFER POOL SIZE				
REGION----	NBA-OBA-USED	REGION----	NBA-OBA-USED	REGION----
1-MDR5	6 0 0	2-MPR5	6 2 0	

- Description:** This display describes the Fast Path buffer pool. It shows its size, the options in effect, and the users of the pool.
- Select Code:** FPBST
- Parameter:** Enter the 1- to 3-digit region number. Default is 1. Enter the number of the region that you want displayed first in the detailed region area.
- Field Descriptions:** Each of the fields is shown and described below by display area.

<u>Area 1</u>				
FAST PATH BUFFER POOL STATISTICS				
30 TOTAL BUFFERS (DBBF)				25 FIXED BUFFERS
10 TOTAL PRE-FIXED PROGRAMS (DBFX)				0 BUFFERS IN USE BY PROGRAMS
2, 048 BUFFER SIZE (BSIZ)				1 BUFFERS BEING WRITTEN

- This area shows the defined characteristics of the Fast Path buffer pool. The descriptions are arranged in alphabetical order.
- BUFFER SIZE (BSIZ)**  
The size of each Fast Path buffer as specified in the JCL.
- BUFFERS BEING WRITTEN**  
The number of fixed buffers that are currently being written by an output thread or that are accumulating sequential dependent segments.
- Note:** Because this number includes buffers assigned to SDEPs, this number can be nonzero even if all output threads are idle.
- BUFFERS IN USE BY PROGRAMS**  
The number of fixed buffers that are currently in use in the dependent regions.

#### FIXED BUFFERS

The total number of buffers that are currently page-fixed.

#### TOTAL BUFFERS (DBBF)

The total number of buffers in the pool as specified in the JCL.

#### TOTAL PRE-FIXED BUFFERS (DBFX)

The total number of buffers to be fixed when the first region starts, as specified in the JCL.

<b><u>Area 2</u></b>	
	24 FIXED BUFFERS AVAILABLE
	5 UNFIXED BUFFERS

This area shows buffer availability. The descriptions are arranged in alphabetical order.

#### FIXED BUFFERS AVAILABLE

The number of fixed buffers available for use by the dependent regions.

#### UNFIXED BUFFERS

The number of buffers that are not page fixed and could be used to start up additional dependent regions.

<b><u>Area 3</u></b>	
2 TOTAL OUTPUT THREADS (OTHR)	2 IDLE OUTPUT THREADS
61,440 TOTAL FAST PATH BUFFER POOL SIZE	
51,200 TOTAL PAGE-FIXED BUFFER POOL SIZE	

This area shows output thread and pool size statistics. The descriptions are arranged in alphabetical order.

#### IDLE OUTPUT THREADS

The number of output threads available for use by the dependent regions.

#### TOTAL FAST PATH BUFFER POOL SIZE

The total virtual storage size of the Fast Path buffer pool (DBBF x BSIZ).

#### TOTAL OUTPUT THREADS (OTHR)

The total number of output threads as specified in the JCL.

## TOTAL PAGE-FIXED BUFFER POOL SIZE

The amount of the Fast Path buffer pool that has been page-fixed by IMS (depends on DBFX, the number of active regions, and the number of open DEDBs).

<b>Area 4</b>											
REGION - - - NBA - OBA - USED				REGION - - - NBA - OBA - USED				REGION - - - NBA - OBA - USED			
1-MDR5	6	0	0	2-MPR5	6	2	0				

This area shows fixed buffer usage by region.

### NBA

The normal buffer allocation specified in the JCL.

### OBA

The overflow buffer allocation specified in the JCL.

### REGION

The region identification and job name.

**Note:** The CICS job name is shown as the message region name when DBCTL threads allocate PSBs with Fast Path resources. The CICS job name appears only while the PSB is allocated. When the transaction completes and the PSB is deallocated, the Fast Path buffers are freed and the CICS job name no longer appears in the display (no DBCTL thread).

### USED

The number of fixed buffers currently in use by the region.

# VSST - VSAM Global Pool Statistics

BMC SOFTWARE			VSAM POOLS			PERFORMANCE MGMT							
SERV ==> VSST			INPUT 13: 57: 26			INTVL=> 3 LOG=> N TGT==> IMSxxx							
PARM ==>			LINE 1 OF 12			SCROLL => CSR							
						<< EXPAND >>							
- - - - - VSAM DATABASE POOL STATISTICS - - - - -													
POOL IS LOCAL			84K TOTAL VSAM POOL SIZE										
DL/I TRACE IS ON			4K TOTAL BUFFER HANDLER POOL SIZE										
BACKGROUND WRITE: 34%			2 BACKGROUND WRITES										
72.4 HIT RATIO FOR ALL SUBPOOLS			5 PURGE/CHKPT CALLS RECEIVED										
- - - - - SUBPOOL SUMMARY - - - - -													
		BUFFER		HIPERSPACE		HIT		READS		WRITES		WRITES	
SP	ID	T	SIZE	BUFFERS	BUFFERS	RATIO	WITH I/O	(USER)		(VSAM)			
1	I001	D	1,024	8	0	98.9	102	0		0			
2	I001	D	2,048	8	0	34.0	340	24		2			
3	I001	D	4,096	15	5	64.1	51	5		0			
4	I002	I	4,096	15	10	92.6	55	1		0			
***** END OF DATA *****													

- Description:** This display describes the VSAM buffer pool: its size, the options in effect, and the defined subpools.
- Select Code:** VSST (global pool statistics)
- Parameter:** None for this global display.
- Entry of a valid subpool number returns the subpool display (see “VSST - VSAM Subpool Statistics” on page 229).
- Expand:** The subpool detail display can be accessed by moving the cursor to the line in the Subpool Summary for the desired subpool and pressing ENTER.
- Scrolling:** The display is scrollable so that any number of subpools can be displayed.
- Field Descriptions:** Each of the fields is shown and described below by display area.

VSAM DATABASE POOL STATISTICS								
POOL IS LOCAL			84K TOTAL VSAM POOL SIZE					
DL/I TRACE IS ON			4K TOTAL BUFFER HANDLER POOL SIZE					
BACKGROUND WRITE: 34%			2 BACKGROUND WRITES					
72.4 HIT RATIO FOR ALL SUBPOOLS			5 PURGE/CHKPT CALLS RECEIVED					

This area shows the defined characteristics of the VSAM database pool. The descriptions are arranged in alphabetical order.

**BACKGROUND WRITE:**

OFF if background write is not selected. If background write is on, this field shows the percent of buffers in each subpool to be considered as candidates for writing by the background write function. This number is specified or defaults to the OPTIONS statement.

**BACKGROUND WRITES**

The number of times background write was invoked since IMS restart.

**DL/I TRACE IS ON/OFF**

The status of the DL/I buffer handler trace is shown as either ON or OFF.

**HIT RATIO FOR ALL SUBPOOLS**

A measure of buffer pool performance for all the VSAM subpools. A high hit ratio means that a high percentage of reads were satisfied from the buffer pools without having to access external storage.

The hit ratio for all subpools is calculated as:

$$100 \times (A / (A + B))$$

where:

A Number of read requests satisfied from all VSAM pools

B Number of reads with I/O for all VSAM pools

**POOL IS LOCAL/GLOBAL**

The pool is identified as either GLOBAL (in CSA) or LOCAL (in IMS private storage).

**PURGE/CHKPT CALLS RECEIVED**

The number of purge or checkpoint calls received since IMS restart.

**TOTAL BUFFER HANDLER POOL SIZE**

Kilobytes of storage for the VSAM statistics, RPL blocks, and the DL/I trace table if tracing is requested.

**TOTAL VSAM POOL SIZE**

The number of kilobytes of storage allocated for all subpools.

<b>Area 2</b>									
----- SUBPOOL SUMMARY -----									
SP	ID	T	BUFFER SIZE	BUFFERS	HIPERSPACE BUFFERS	HIT RATIO	READS WITH I/O	WRITES (USER)	WRITES (VSAM)
1	I001	D	1,024	8	0	98.9	102	0	0
2	I001	D	2,048	8	0	34.0	340	24	2
3	I001	D	4,096	15	5	64.1	51	5	0
4	I002	I	4,096	15	10	92.6	55	1	0

This area shows the characteristics of each defined subpool and the I/O activity since IMS restart. The descriptions are arranged in alphabetical order.

#### **BUFFERS**

Number of virtual storage buffers for this subpool.

#### **BUFFER SIZE**

The size of a subpool buffer.

#### **HIPERSPACE BUFFERS**

Number of Hiperspace buffers defined for this subpool.

#### **HIT RATIO**

A measure of buffer pool performance. A high hit ratio means that a high percentage of reads are satisfied from the buffer pool without having to access external storage.

The hit ratio for each subpool is computed as follows:

$$100 \times (A / (A + B))$$

where:

A        Requests satisfied from pool

B        VSAM reads with I/O

#### **ID**

Four character pool ID for this VSAM subpool: xxxx is shown if an ID is if not specified.

#### **READS WITH I/O**

Represents the number of times VSAM had to access external storage because the VSAM control interval was not in the virtual storage or Hiperspace buffer pool.

#### **SP**

Relative subpool number.

#### **T**

Subpool type. It can be:

D	Data
I	Index

#### **WRITES (USER)**

Writes by VSAM initiated by the user for synchronization point processing.

#### **WRITES (VSAM)**

Number of writes initiated by VSAM to make free space in the pool. These types of writes are very expensive and indicate that the pool may be too small. They also are called VSAM steal writes or forced writes.

## VSST - VSAM Subpool Statistics

BMC SOFTWARE -----			VSAM POOLS -----			PERFORMANCE MGMT		
SERV ==>	VSST		INPUT	13: 57: 59	INTVL=> 3	LOG=> N	TGT==> IMSxxx	
PARM ==>	04				LINE	1 OF	20	SCROLL=> CSR
----- IMS VSAM SUBPOOL STATISTICS -----								
SP	4	BUFFER SIZE	4,096		15 BUFFERS		PAGEFIX	BUFFERS N
ID	I001	BFR TYPE	INDEX		10 H/S BUFFERS		PAGEFIX	IO BLKS N
----- IMS VSAM BUFFER HANDLER REQUESTS -----								
		0 RETRIEVES BY RBA			0 ESDS INSERTS			
		500 RETRIEVES BY KEY			1 KSDS INSERTS			
					0 ALTERS			
----- VSAM RETRIEVE REQUESTS / VSAM PERFORMANCE STATISTICS -----								
		500 VSAM GETS			0 WRITES (VSAM INITIATED)			
		0 VSAM SEARCH BUFFER POOL			1 WRITES (USER INITIATED)			
		691 SATISFIED FROM POOL						
		55 READS WITH I/O			0 ERROR BUFFERS IN POOL			
		92.6 HIT RATIO			0 MAX ERROR BUFFERS IN POOL			
----- HIPERSPACE BUFFER STATISTICS -----								
		42.9 HIPERSPACE HIT RATIO						
		320 SUCCESSFUL READS			6 SUCCESSFUL WRITES			
		0 UNSUCCESSFUL READS			0 UNSUCCESSFUL WRITES			
***** END OF DATA *****								

**Description:** For the VSAM buffer pool, most of the statistics are kept by subpool. This display shows the requests and work performed in the selected subpool since IMS restart.

This display can be accessed by cursor selection from the VSST global display or by parameter specification.

**Select Code:** VSST (subpool statistics)

**Parameter:** Enter subpool number (1 or 2 digits).

If no entry is made, the global pool display (see “VSST - VSAM Global Pool Statistics” on page 226) is returned.

**Scrolling:** This display is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<u>Area 1</u>								
----- IMS VSAM SUBPOOL STATISTICS -----								
SP	4	BUFFER SIZE	4,096		15 BUFFERS		PAGEFIX	BUFFERS N
ID	I001	BFR TYPE	INDEX		10 H/S BUFFERS		PAGEFIX	IO BLKS N

This area shows the current subpool definition. The descriptions are arranged in alphabetical order.

**BFR TYPE**

Possible values:

**DATA**

Data buffers and any index buffers not assigned to an index pool in this pool (default)

**INDEX**

Only index buffers in this pool

**BUFFERS**

The number of buffers specified.

**BUFFER SIZE**

The subpool buffer size in bytes.

**H/S BUFFERS**

Number of Hipspace buffers specified for this subpool.

**ID**

Four-character VSAM subpool ID.

**PAGEFIX BUFFERS**

Indicates (Y/N) whether buffers are page fixed for this subpool.

**PAGEFIX IO BLKS**

Indicates (Y/N) whether I/O blocks are page fixed for this subpool.

**SP**

Relative subpool number.

<b><u>Area 2</u></b>	
----- IMS VSAM BUFFER HANDLER REQUESTS -----	
0 RETRIEVES BY RBA	0 ESDS INSERTS
500 RETRIEVES BY KEY	1 KSDS INSERTS
	0 ALTERS

This area displays the number of requests made to this subpool, separated by the type of request. The descriptions are arranged in alphabetical order.

**ALTERS**

The number of alters.

**ESDS INSERTS**

The number of ESDS inserts.

**KSDS INSERTS**

The number of KSDS inserts.

## RETRIEVES BY KEY

The number of retrieves by key.

## RETRIEVES BY RBA

The number of retrieves by RBA.

### Area 3

```
- - - - - VSAM RETRIEVE REQUESTS / VSAM PERFORMANCE STATISTICS - - - - -
      500 VSAM GETS                                0 WRITES (VSAM INITIATED)
        0 VSAM SEARCH BUFFER POOL                  1 WRITES (USER INITIATED)
      691 SATISFIED FROM POOL
        55 READS WITH I/O                          0 ERROR BUFFERS IN POOL
      92.6 HIT RATIO                               0 MAX ERROR BUFFERS IN POOL
```

This area displays the retrieve requests made to VSAM and the amount of VSAM I/O required to satisfy all requests. The descriptions are arranged in alphabetical order.

## ERROR BUFFERS IN POOL

Error buffers currently in the pool.

## HIT RATIO

A measure of buffer pool performance. A high hit ratio means that a high percentage of reads are satisfied from the buffer pool without having to access external storage.

The hit ratio is computed as follows:

$$100 \times (A / (A + B))$$

where:

A        Requests satisfied from pool

B        VSAM reads with I/O

## MAX ERROR BUFFERS IN POOL

The largest number of error buffers ever in the pool since IMS restart.

## READS WITH I/O

Represents the number of times VSAM had to access external storage because the VSAM control interval was not in the virtual storage or Hiperspace buffer pool.

## SATISFIED FROM POOL

Requests satisfied with a record already in the pool.

## VSAM GETS

The number of VSAM GET requests.

## VSAM SEARCH BUFFER POOL

The number of VSAM search buffer requests.

#### WRITES (USER INITIATED)

Writes by VSAM initiated by the user for synchronization point or background write processing.

#### WRITES (VSAM INITIATED)

Writes initiated by VSAM to free space in the pool. These writes are very expensive and indicate that the pool may be too small. They are also called VSAM steal writes or forced writes.

<b>Area 4</b>	
----- HIPERSPACE BUFFER STATISTICS -----	
42.9 HIPERSPACE HIT RATIO	
320 SUCCESSFUL READS	6 SUCCESSFUL WRITES
0 UNSUCCESSFUL READS	0 UNSUCCESSFUL WRITES

This area shows the number of Hiperspace reads and writes requested. The descriptions are arranged in alphabetical order.

#### HIPERSPACE HIT RATIO

The portion of the VSAM hit ratio that is due to Hiperspace buffering. It is calculated as follows:

$$100 \times (A / (B + C))$$

where:

- A Successful reads from Hiperspace
- B Requests satisfied from pool (includes successful Hiperspace reads)
- C Reads with I/O

#### SUCCESSFUL READS

The number of successful reads from Hiperspace for this subpool.

#### SUCCESSFUL WRITES

The number of successful writes to Hiperspace for this subpool.

#### UNSUCCESSFUL READS

The number of unsuccessful reads from Hiperspace. Each unsuccessful read represents an occasion when data had to be obtained from external storage because the Hiperspace page was stolen. If IMS determines in advance that the desired CI is not in Hiperspace, it does not attempt the Hiperspace read. Therefore, the number of reads to external storage can be greater than the number of unsuccessful Hiperspace reads.

#### UNSUCCESSFUL WRITES

The number of unsuccessful writes to Hiperspace.

# Chapter 14. IMS Internals Displays

This chapter describes the displays that show the status of IMS resources when internal IMS functions are invoked for transaction processing.

## APPCA - APPC Activity Summary

BMC SOFTWARE				-----				APPC ACTIVITY				-----				PERFORMANCE MGMT																			
SERV ==>		APPCA		INPUT		12: 58: 43		INTVL=>		3		LOG=>		N		TGT==>		IMSA																	
PARAM ==>		ROW																		1		OF		2		SCROLL=>		CSR							
EXPAND:		APPCL,		USER,		LINESEL(DAPPC),		FSEL(+)																											
																				---				ACTIVE CONVERSATIONS				---							
IMS		BASE		LU		NAME:		IMSLU62		CURRENT		STATUS:		ENAB		SYNC:		1		INPUT:		1													
		XRF		LU		NAME:				DESIRED		STATUS:		ENAB		ASYN:		0		OUTPUT:		0													
-----																																			
NETID/				TPNAME				-----																											
LUNAME				D CONVERSATION ID				USERID				QCNT				TRANCODE				CLS				RGN				MSG				STATUS			
-----																																			
												+++++++				---		+++		---															
RXAPPC		*		0		DFSSIDE																													
												0000000000000000				3								STO											
MVSTEST		I		INQUIRY_PART																															
												03C8F62800000012				JCL3				PART				1		1		IMP							
*****																																			
												END OF DATA				*****																			

**Description:** This display shows summary information for the IMS APPC connection and an individual line for each LUNAME/TPNAME currently in a conversation with IMS as one of the partner LUs.

If there is no IMS support for APPC, the following message appears:

IMS DOES NOT SUPPORT APPC

with no other data displayed on the screen.

**Color:** If you have a color monitor:

- Red Highlights the:
- Current APPC connection (CURRENT STATUS) and the requested APPC connection (DESIRED STATUS) if they are different.
  - LU name and TP name rows for LU and TP name combinations that are stopped.

**Select Code:** APPCA

**Parameter:**

All of the APPCA parameters, except for SORT, act as filters that restrict the information shown according to the criteria specified by the parameters.

Use the parameters as follows:

- Multiple parameters must be separated by commas.

If multiple filtering parameters are entered, **all** restrictions must be met.

- A blank indicates the end of a parameter string; it cannot be used as a delimiter.
- Multiple resources with similar names can be requested by using an \* character as a generic qualifier and a + character as a positional qualifier. The positional qualifier is repeated for every character to be replaced. The generic qualifier replaces a group of characters. For example, a parameter of LU=AB+D\* shows all LUs that start with AB, have D in the fourth position, and have any character following D.

Invalid parameter combinations include:

If one parameter invalidates another, the following warning message is issued:

**INCONSISTENT PARAMETERS ENTERED, xxx IGNORED**

where xxx is the parameter ignored; processing continues.

Invalid parameter combinations and the action taken are:

**D=I, QCNT>n | XON**

**Warning Message: D=I PARAMETER IGNORED**

The invalid D parameter is replaced with D=0 and all valid data is displayed.

**D=0, USERID=i d | TRANCODE=t rncode | CLASS=nnn | RGN=nnn**

**Warning Message: D=0 PARAMETER IGNORED**

The invalid D parameter is replaced with D=I and all valid data is displayed.

**QCNT>n | XON, USERID=i d | TRANCODE=t rncode | CLASS=nnn | RGN=nnn**

**Error Message: QCNT AND XON INVALID WITH TRANCODE, USERID, CLASS AND RGN PARMS**

Valid keywords for APPCA parameters include the following. The parameter descriptions are arranged in alphabetical order.

CLASS|CLS=nnn

**Note:** Where nnn is a valid class number. Displays only those entries with a matching class.

When this parameter is entered, D=I is assumed since only input allocation directions have a class.

CONVID=hex

Where hex must be a 16-character hexadecimal ID. Displays only those conversation IDs that match the CONVID value.

D=I|O

Displays only those LUNAME/TPNAME combinations that match the allocation direction request where:

I Allocation requests directed to IMS as input.

O Allocation requests directed from IMS as output.

LU=name

Where name is a 1- to 8-character LU name. Generic and positional qualifiers can be used for the LU name. Displays only those LU names that have a matching name.

QCNT|Q>n

Where n can be a 1- to 4-character count. Displays only those LUNAME/TPNAME entries with a queue count greater than n.

**Note:** When this parameter is entered, D=O is assumed since only output allocation directions have a queue count.

RGN=nnn

Where nnn is a valid region number. Displays only those entries with a matching region number.

**Note:** When this parameter is entered, D=I is assumed since only input allocation directions have a region.

SORT|SO=cc

Where cc can be any of the following two characters. The display list is sorted by QCNT in descending numerical order by default. The primary sort is by a selected column. A secondary sort for all columns is by LUNAME and TPNAME in ascending alphabetical order.

The following descriptions are arranged in alphabetical order:

- |    |   |
|----|---|
| CL | Sorts the list in descending numerical order by CLS (class).  |
| CO | Sorts the list in descending numerical order by CONVERSATION ID.                                    |
| D  | Sorts the list alphabetically by D (allocation direction).  |
| LU | Sorts the list alphabetically by LUNAME.<br><br>This value includes the LU summary line.            |
| MS | Sorts the list alphabetically by MSG (input message type).  |
| NE | Sorts the list alphabetically by network ID of the LU name  |
| QC | Sorts the list in descending numerical order by QCNT (queue count) with a secondary sort on LUNAME. |
| RG | Sorts the list in descending numerical order by RGN (region).                                       |
| ST | Sorts the list alphabetically by STATUS.  |
| TP | Sorts the list alphabetically by TPNAME.  |
| TR | Sorts the list alphabetically by TRANCODE.  |
| US | Sorts the list alphabetically by USERID.  |

TP=name

Where **name** is a TP name (maximum of 52 characters). Generic and positional qualifiers can be used for the TP name. Displays only those TP names that have a matching name.

TRANCODE|TRAN|TR=trancode

Where **trancode** can be a 1- to 8-character transaction name. Generic and positional qualifiers can be used for the transaction name. Displays only those transaction codes that have a matching name.

**Note:** When this parameter is entered, D=I is assumed since only input allocation directions have a transaction code.

USERID|US=i d

Where i d is a 1- to 8-character user ID. Generic and positional qualifiers can be used for the user ID. Displays only those users that have a matching ID.

**Note:** When this parameter is entered, D=I is assumed since only input allocation directions have a user ID.

XON

Displays only LUNAME/TPNAME entries when there is a status in the STATUS field.

**Expand:**

APPC can be EXPANDED by using the following fields as indicated:

APPCL

Move the cursor to this field and press ENTER to invoke the APPC LU status display.

USER

Move the cursor to this field and press ENTER to invoke the user status summary display (USER).

LINESEL(DAPPC)

Move the cursor to an LU name in one of the rows of the APPCA display and press ENTER to invoke the DAPPC service. DAPPC shows detail information for the selected LU name.

Selecting DAPPC with the cursor from the EXPAND line invokes the DAPPC service for the LU name in the first row of the APPCA display.

FSEL(+)

Indicates that at least one column has cursor-sensitive fields. A string of + characters underneath a column name means that this column contains cursor-sensitive fields. Moving the cursor to a field in that column and pressing the ENTER key invokes a related display for that field. For example, if the cursor is at a field value under the RGN column, pressing the ENTER key displays the REGNS service with the correct region number as the parameter. Columns with cursor-sensitive fields include:

RGN

The REGNS service showing IMS dependent region activity for the transaction currently processing can be displayed with the region number as a parameter.

## TRANCODE

The TRANQ service showing transaction status details can be displayed with the transaction code as a parameter.

### Sorting:

The display list can be sorted by:

- Using the SORT parameter.
- Positioning the cursor with the TAB key to the column heading to be sorted and pressing ENTER.

This overrides any SORT parameter entered in the PARM field and primes the field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by QCNT by default with a secondary sort by LUNAME and TPNAME.

### Scrolling:

The display is scrollable.

### Field Descriptions:

Each of the fields is shown and described below by display area.

<b><u>Area 1</u></b>							
				--- ACTIVE CONVERSATIONS ---			
IMS BASE LU NAME: IMSLU62	CURRENT STATUS: ENAB	SYNC:	1	INPUT:	1		
XRF LU NAME:	DESIRED STATUS: ENAB	ASYNC:	0	OUTPUT:	0		

This area shows APPC status. The descriptions are arranged in alphabetical order.

#### ASYNC

The count of currently active asynchronous conversations.

#### CURRENT STATUS

**Note:** If the DESIRED STATUS is different from the CURRENT STATUS, both fields are highlighted in red.

The current status of the IMS connection to APPC. This can be:

CAN	The APPC/IMS connection was cancelled.
DISA	The APPC/IMS connection is disabled.
ENAB	The APPC/IMS connection is enabled (active).
FAIL	The APPC/IMS connection failed.
OUTB	The APPC/IMS connection is outbound.
PURG	The APPC/IMS connection is purging.
STOP	The APPC/IMS connection stopped.

STRT            The APPC/IMS connection is starting.

#### DESIRED STATUS

The requested status of the IMS connection to APPC. This can be:

CAN            The APPC/IMS connection was cancelled.

DISA           The APPC/IMS connection is disabled.

ENAB           The APPC/IMS connection is enabled (active).

FAIL           The APPC/IMS connection failed.

OUTB           The APPC/IMS connection is outbound.

PURG           The APPC/IMS connection is purging.

STOP           The APPC/IMS connection stopped.

STRT           The APPC/IMS connection is starting.

#### IMS BASE LU NAME

The LU name used by APPC to route connections to this IMS subsystem. This value is defined in SYS1.PARMLIB member APPCPMxx where xx is a user-defined APPC startup parameter. This field contains N/A when IMS is not connected to APPC.

#### INPUT

The count of currently active input conversations.

#### OUTPUT

The count of currently active output conversations.

#### XRF LUNAME

The LU name of the XRF system associated with this IMS.

**Note:** Neither this field nor its label is displayed if XRF is not defined to the system.

#### SYNC

The count of currently active synchronous conversations.

#### Area 2

NETID/ LUNAME	TPNAME	-----	D	CONVERSATION ID	USERID	QCNT	TRANCODE	CLS	RGN	MSG	STATUS
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
USBMC	*	0	DFSSIDE								
RXAPPC			0000000000000000			3					STO
USBMC	I		INQUIRY_PART								
MVSTEST			03C8F62800000012	JCL3			PART	1	1	IMP	

This area contains one entry for every LU name, TP name, and conversation connection to IMS. The descriptions are arranged in alphabetical order.

#### CLS

The IMS scheduling class of this transaction.

**Note:** This field shows a class only when the value in the D column is I.

#### CONVERSATION ID

The conversation ID for this conversation. For output conversation, it might be zeros.

#### D

The direction of the allocation request. This can be:

- I This allocation request is directed as input to IMS from the LU name. The conversation is allocated by the partner LU as shown in the LUNAME column.
- O This allocation request is directed as output to the LU name from IMS. The conversation is allocated by IMS.

#### MSG

Type of message received from this TP name:

- CMD IMS command.
- CNV Conversational transaction.
- EMH Expedited Message Handler (Fast Path) buffered transaction.
- EXP Explicit transaction (scheduling information derived from a TP name definition in APPC).
- IMP Implicit transaction (scheduling information derived from a TRANSACT macro).
- MSW Message switch.
- REC Recoverable transaction.
- RSP Transaction response.

**Note:** This field shows a message type only when the value in the D column is I.

#### NETID/LUNAME

The network ID of the LU name or the LU name of the partner LU to IMS for this conversation. Depending on how the conversation is initiated, this value is often not an actual LU name when the conversation is initiated by IMS (the value in the D column is O); for example:

- The LU name is actually the name of a SIDEINFO entry in APPC. This results from an allocation request specifying only the SIDEINFO name with no other conversation parameters.

- The LU name is DFSLU, which is the default LU name used by IMS when no LU name is supplied.

An asterisk indicates that this LU name is actually a SIDEINFO name.

#### QCNT

Count of requests currently queued to this LU name and TP name awaiting execution.

**Note:** This field shows a count only when the value in the D column is O.

#### RGN

The region where this transaction is executing.

**Note:** This field shows a value only when the D column contains an I. If 0 is shown, this means the region is not scheduled.

#### STATUS

The following statuses (up to a maximum of 10 characters) can appear in this field in the order shown below. To see all the statuses associated with an LUNAME/TPNAME, use the DAPPC service. To see all the statuses associated with an LUNAME only, use the APPCL service.

**STO** This LUNAME/TPNAME is stopped. If this is the status for the LUNAME/TPNAME, this line is highlighted in red.

**LUSTO** This LUNAME is stopped. The LUNAME/TPNAME may still be active, but the LUNAME is stopped. If this is the status for the LU, this line is highlighted in red.

**TRA** Trace is on for this LUNAME/TPNAME.

**LUTR** Trace is on for this LUNAME only.

#### TPNAME

The TP name used in the initiation of the conversation by APPC with IMS to control scheduling.

- If this is an input conversation, this TPNAME is passed to APPC by the partner LU as shown in the LUNAME field.
- If this is an output TP name, then this TPNAME is passed to APPC by IMS.

#### TRANCODE

The transaction code scheduled for this TP name. This is specified in the APPC TP name definition and may be a transaction code specified in a TRANSACT macro or an implicit APPC transaction, or it may be the name of a program associated with an explicit APPC transaction.

**Note:** This field shows a transaction code only when the value in the D column is I.

#### USERID

The ID of the user who established the connection. For input connections, this is the ID of the user currently connected to this LU with this TP name.

**Note:** This field shows an ID only when the value in the D column is I.

# APPCL - APPC LU Status

BMC SOFTWARE -----			APPC LU STATUS			-----PERFORMANCE MGMT					
SERV ==> APPCL			INPUT 07: 21: 13			INTVL=> 3					
LOG=> N			TGT==> IMSA								
PARAM ==>			ROW 1 OF 6			SCROLL=> CSR					
EXPAND: APPCA, LI NESEL(APPCA)											
			ACTIVE CONVERSATIONS								
IMS BASE LU NAME: IMSLU62			CURRENT STATUS	ENAB	SYNC :	1	INPUT:	1			
XRF LU NAME:			DESIRED STATUS	ENAB	ASYN:	1	OUTPUT:	1			
NETID	LUNAME	DIRECT	CONV	QCNT	ENQCT	TPCNT	TPSTP	STATUS			
-----											
USBMC	MVSTEST	INPUT	1			1		TRA			
USBMC	MVSTEST	OUTPUT	0	0	0	0	0				
USBMC	RXAPPC	INPUT	0			0					
USBMC	RXAPPC *	OUTPUT	1	6	6	1	0				
USBMC	RXAPPC2	INPUT	0			0					
USBMC	RXAPPC2 *	OUTPUT	0	1	1	1	0				
***** END OF DATA *****											

**Description:** This display shows summary information for the IMS APPC connection and an individual line for each LU in a conversation with IMS as one of the partner LUs.

If IMS is not connected to APPC, the following message appears:

IMS NOT CONNECTED TO APPC

with no other data displayed on the screen.

**Color:** If you have a color monitor:

Red Highlights the:

- Current APPC connection (CURRENT STATUS) and the requested APPC connection (DESIRED STATUS) if they are different.
- LU names that are stopped or have a nonzero TPSTP value.
- LU names with a status of STO.

**Select Code:** APPCL

**Parameter:** All of the APPCL parameters, except for SORT, act as filters that restrict the information shown according to the criteria specified by the parameters.

Use the parameters as follows:

- Multiple parameters must be separated by commas.

If multiple filtering parameters are entered, all restrictions must be met.

- A blank indicates the end of a parameter string; it cannot be used as a delimiter.
- Multiple resources with similar names can be requested by using an \* character as a generic qualifier and a + character as a positional qualifier. The positional qualifier is repeated for every character to be replaced. The generic qualifier replaces a group of characters. For example, a parameter of LU=AB+D\* shows all LUs that start with AB, have D in the fourth position, and have any character following D.

Invalid parameter combinations include:

If one parameter invalidates another, the following warning message is issued:

**INCONSISTENT PARAMETERS ENTERED, xxx IGNORED**

where xxx is the parameter ignored; processing continues.

Invalid parameter combinations and the action taken are:

**D=I, QCNT>n | XON**

**Warning Message: D=I PARAMETER IGNORED**

The invalid D parameter is replaced with D=0.

Valid keywords for APPCL parameters include:

**DIRECT|D=I|O**

Displays only those LU names that match the allocation direction request where:

**I** Allocation requests directed to IMS as input.

**O** Allocation requests directed to IMS as output.

**LU=name**

Where **name** is a 1- to 8-character LU name. Generic and positional qualifiers can be used for the LU name. Displays only those LU names that have a matching name.

**QCNT|Q>n**

Where **n** can be a 1- to 4-character count. Displays only those LUNAME entries with a queue count greater than **n**.

**Note:** When this parameter is entered, D=0 is assumed since only output allocation directions have a queue count.

`SORT|SO=cc`

Where `cc` can be any of the following two characters. The display list is sorted by `QCNT` in descending numerical order by default. The primary sort is by a selected column. A secondary sort for all columns is by `LUNAME` in ascending alphabetical order.

**CO** Sorts the list in descending numerical order by `CONV` (count of active conversations).

This value includes the LU summary line.

**DI** Sorts the list alphabetically by `DIRECT` (allocation direction).

**EN** Sorts the list in descending numerical order by the `ENQCT` (enqueue count).

**LU** Sorts the list alphabetically by `LUNAME`.

This includes the LU summary line.

**NE** Sorts the list alphabetically by the network ID of the LU name.

**QC** Sorts the list in descending numerical order by `QCNT` (queue count) by default with a secondary sort on `LUNAME`.

**ST** Sorts the list alphabetically by `STATUS`.

**TP** Sorts the list in descending numerical order by `TPCNT` (count of TP names for an LU name).

**TS** Sorts the list in descending numerical order by `TPSTP` (count of stopped TPs for an LU name).

**XON**

Displays only `LUNAME` entries when there is a status in the `STATUS` field or when `TPSTP` contains a nonzero value.

**Expand:**

`APPCL` can be EXPANDED by moving the cursor to the following fields in the `EXPAND` line and pressing `ENTER`:

**APPCA** APPC activity summary display.

**LINSEL(APPCA)** APPCA activity summary display for the first LU name in `LUNAME` of the `APPCL` display.  
The `PARM` field of the `APPCA` display is primed with the LU name.

You can also move the cursor to a field in the `LUNAME` column and press `ENTER` to `EXPAND` to the selected LU in the `APPCA` display.

**Sorting:**

The display list can be sorted by:

- Using the SORT parameter.
- Positioning the cursor with the TAB key to the column heading to be sorted and pressing ENTER.

This overrides any SORT parameter entered in the PARM field and primes the field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by QCNT by default with a secondary sort by LUNAME.

**Scrolling:**

The display is scrollable.

**Field Descriptions:**

Each of the fields is shown and described below by display area.

<b><u>Area 1</u></b>							
				ACTIVE CONVERSATIONS			
IMS BASE LU NAME: IMSLU62	CURRENT STATUS	ENAB	SYNC :	1	INPUT:	1	
XRF LU NAME:	DESI RED STATUS	ENAB	ASYN:	1	OUTPUT:	1	

This area shows APPC status. The descriptions are arranged in alphabetical order.

**ASYN**

The count of currently active asynchronous conversations.

**CURRENT STATUS**

**Note:** If the DESIRED STATUS is different from the CURRENT STATUS, both fields are highlighted in red.

The current status of the IMS connection to APPC. This can be:

CAN	The APPC/IMS connection was cancelled.
DISA	The APPC/IMS connection is disabled.
ENAB	The APPC/IMS connection is enabled (active).
FAIL	The APPC/IMS connection failed.
OUTB	The APPC/IMS connection is outbound.
PURG	The APPC/IMS connection is purging.
STOP	The APPC/IMS connection stopped.
STRT	The APPC/IMS connection is starting.

**DESIRED STATUS**

The requested status of the IMS connection to APPC. This can be:

CAN	The APPC/IMS connection was cancelled.
DISA	The APPC/IMS connection is disabled.

ENAB      The APPC/IMS connection is enabled (active).

FAIL      The APPC/IMS connection failed.

OUTB      The APPC/IMS connection is outbound.

PURG      The APPC/IMS connection is purging.

STOP      The APPC/IMS connection stopped.

STRT      The APPC/IMS connection is starting.

#### IMS BASE LU NAME

The LU name used by APPC to route connections to this IMS subsystem. This is defined in SYS1.PARMLIB member APPCPMxx where xx is a user-defined APPC startup parameter. This field contains N/A when IMS is not connected to APPC.

#### INPUT

The count of currently active input conversations.

#### OUTPUT

The count of currently active output conversations.

#### SYNC

The count of currently active synchronous conversations.

#### XRF LUNAME

The LU name of the XRF system associated with this IMS.

**Note:** Neither this field nor its label is displayed if XRF is not defined to the system.

<b>Area 2</b>								
NETID	LUNAME	DIRECT	CONV	QCNT	ENQCT	TPCNT	TPSTP	STATUS
-----	-----	-----	-----	-----	-----	-----	-----	-----
USBMC	MVSTEST	INPUT	1			1		TRA
USBMC	MVSTEST	OUTPUT	0	0	0	0	0	
USBMC	RXAPPC	INPUT	0			0		
USBMC	RXAPPC *	OUTPUT	1	6	6	1	0	
USBMC	RXAPPC2	INPUT	0			0		
USBMC	RXAPPC2 *	OUTPUT	0	1	1	1	0	

This area contains a one-line summary entry for each LU allocation direction by LU name. The descriptions are arranged in alphabetical order.

#### CONV

The current number of active conversations. For output conversation, it might be zeros.

#### DIRECT

The direction of the allocation request. This can be:

- INPUT      This allocation request is directed as input to IMS from the LU name. The conversation is allocated by the partner LU as shown in the LUNAME column.
- OUTPUT     This allocation request is directed as output to the LU name from IMS. The conversation is allocated by IMS.

#### ENQCT

Count of requests queued to all TPs for this LU name since the last IMS cold start.

This field contains a count only when the value in the DIRECT column is OUTPUT.

#### LUNAME

The LU name of the partner LU to IMS for this conversation. Depending on how the conversation is initiated, this is often not an actual LU name when the conversation is initiated by IMS (the value in the DIRECT column is OUTPUT); for example:

- The LU name is actually the name of a SIDEINFO entry in APPC. This results from an allocation request specifying only the SIDEINFO name with no other conversation parameters.
- The LU name is DFSLU. This is the default LU name used by IMS when no LU name is supplied.

An asterisk indicates that this LU name is actually a SIDEINFO name.

#### NETID

The network ID of the LU name.

#### QCNT

Count of requests that are awaiting execution and are currently queued to all TPs for this LU name.

**Note:** This field shows a count only when the value in the DIRECT column is OUTPUT.

#### STATUS

The status of this LU name:

STO      This LU name is stopped. If this is the status for the LU, this line is highlighted in red.

TRA      The trace is set on for this LU name.

#### Notes:

- If the LU name is stopped, this status appears as the left-most status.
- This field shows a status only when the value in the DIRECT column is OUTPUT.

#### TPCNT

The count of all TP names for this LU name.

- For OUTPUT allocation direction, this is a count of TP names since the last IMS cold start.
- For INPUT allocation direction, this is a count of all currently active TP names.

#### TPSTP

A count of the number of TP names that are stopped for this LU name.

# DAPPC - Input Allocation Direction

```
BMC SOFTWARE ----- APPC DETAIL ----- PERFORMANCE MGMT
SERV ==> DAPPC   INPUT   07:24:43  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==> CONVID=03C8F62800000012      ROW    1 OF   12  SCROLL=> CSR
EXPAND: APPCL, APPCA, DREGN, TRANQ

LU NAME : MVSTEST      NETID: USBMC
ACTIVE CONV:      1                LU STATUS: TRA

TPNAME : INQUIRY_PART

CONVID : 03C8F62800000012
DIRECTION : INPUT          TRANCODE: PART
MODENAME : LU62APPC        TRANTYPE: IMPLICIT
MSG      : IMP             CLASS   : 1
CONV TYPE: MAPPED          QUEUED  : 0
SYNC LEVEL: CONFIRM        RGN #   : 1
USERID   : JCL3            RGNNAME : DFSMPR
                                PSBNAME: DFSSAM02
                                DB2STAT: N/A
                                DLI TOT: 2
                                SQL TOT: 0
***** END OF DATA *****
```

**Description:** This display can be selected by EXPANDING from the APPCA display or by selecting DAPPC from the analyzer service list (Primary Option Menu 1). If you select this service from the list and do not specify a conversation ID, the first conversation found is displayed.

The DAPPC display shown is determined by the allocation direction of the conversation. The display for the input allocation direction shows detail information about an inbound allocation conversation to IMS. It shows the LU, TP, and the conversation. The data displayed for the input allocation direction TP names reflects the information passed to IMS from APPC.

If the IMS control region supports APPC but there is no APPC connection, any data available for this conversation and the following message are displayed:

IMS NOT CONNECTED TO APPC

If the IMS control region does not support APPC, the following message is displayed only:

IMS DOES NOT SUPPORT APPC

**Color:** If you have a color monitor:

- Red Highlights the:
- LU STATUS field if the LU is stopped.
  - TP STATUS field if the TP is stopped.

**Select Code:** DAPPC

**Parameter:** CONVID=hex

Where hex is a 16-character hexadecimal ID.

If this parameter is not specified, the first conversation found is displayed. The conversation could be in an output allocation direction instead of an input allocation direction, depending upon the conversation found.

**Expand:** DAPPC can be EXPANDED by moving the cursor to the following fields in the EXPAND line and pressing ENTER:

APPCL      APPC LU status display.

APPCA      APPCA activity summary display.

DREGN      Region detail display.

The PARM field of the DREGN display is primed with the RGNNAME value from DAPPC.

TRANQ      Transaction queue status display.

The PARM field of the TRANQ display is primed with the TRANCODE value from DAPPC.

**Scrolling:** The display is scrollable below TPNAME.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>			
LU NAME :	MVSTEST	NETID:	USBMC
ACTIVE CONV:	1	LU STATUS:	TRA

This area shows the LU name and its status. The descriptions are arranged in alphabetical order.

#### ACTIVE CONV

The count of currently active conversations for this LU name in the allocation direction shown by the DIRECTION field.

#### LU NAME

The LU name of the partner LU to IMS for this conversation. Depending on how the conversation is initiated, this is often not an actual LU name when the conversation is initiated by IMS (the value in the DIRECTION field is OUTPUT); for example:

The LU name is DFSLU. This is the default LU name used by IMS when no LU name is supplied.

## LU STATUS

The status of this LU:

**STO** This LU name is stopped. The field is highlighted in red.

**TRA** Trace is on for this LU name.

**Note:** If the LU name is stopped, this status appears as the left-most status.

## NETID

The network ID of the LU name.

### Area 2

TPNAME : INQUIRY\_PART

This area shows the TP name.

## TPNAME

The TP name used in the initiation of the conversation by the partner LU/TP to IMS LU.

### Area 3

CONVID : 03C8F62800000012  
DIRECTION : INPUT  
MODENAME : LU62APPC  
MSG : IMP  
CONV TYPE : MAPPED  
SYNC LEVEL: CONFIRM  
USERID : JCL3

This area shows conversation-level data. The descriptions are arranged in alphabetical order.

## CONVID

The conversation ID used in this APPC conversation.

## CONV TYPE

Conversation type:

**BASIC** Basic conversation.

**MAPPED** Mapped conversation.

## DIRECTION

The direction of the allocation request as INPUT.

This allocation request is directed as input to IMS from the LU name. The conversation is allocated by the partner LU as shown in the LU NAME field.

## MSG

Type of message received from this TP name:

CMD	IMS command.
CNV	Conversational transaction.
EMH	Expedited Message Handler (Fast Path) buffered transaction.
EXP	Explicit transaction.
	Scheduling information is derived from a TP name definition in APPC.
IMP	Implicit transaction.
	Scheduling information is derived from a TRANSACT macro.
MSW	Message switch.
REC	Recoverable transaction.
RSP	Transaction response.

## MODENAME

The mode name used by the partner LU/TP to allocate this conversation.

## SYNC LEVEL

Synchronization level:

CONFIRM	Confirmation is required for CPI-C SEND requests.
NONE	No synchronization.

## USERID

The ID of the user who established the connection:

- For output connections, this is the user ID of the last user to connect to this LU with this TP name.
- For input connections, this is the user ID of the user currently connected to this LU with this TP name.

### Area 4

```
TRANCODE: PART
TRANSTYPE: IMPLICIT
CLASS    :    1
QUEUED   :    0
RGN #    :    1
RGNNAME  : DFSMPR
PSBNAME  : DFSSAM02
DB2STAT  : N/A
DLI TOT  :    2
SQL TOT  :    0
```

This area shows IMS-related data. The descriptions are arranged in alphabetical order.

## CLASS

Processing class for this transaction.

## DB2STAT

The DB2 connection status for the region. If the region is connected, the first half of the DB2 status message shows the DB2 subsystem name. The second half of the message shows the connection status:

- CON           IMS connects this region to DB2 if a connection is available and the EXEC parameter SSM (to establish a DB2 connection) is valid.
- SON           The application is signed on and a recovery token is assigned by DB2 if the EXEC parameter SSM is valid, the connection is successful, and the application issues a DB2 request.
- THD           IMS had a thread with DB2 when it processed the EXEC parameter SSM to establish a connection to DB2.

If a region is connected to more than one DB2 subsystem and a thread is active, -THD is shown. If no thread is active, only the first connection is shown.

## DLI TOT

The total number of DL/I calls since program scheduling.

## PSBNAME

The PSB name (if any) currently scheduled for this transaction.

## QUEUED

Number of currently queued input messages with this transaction code.

## RGN #

The region where this transaction is currently executing.

## RGNNAME

The name of the region where this transaction is currently executing.

**Note:** If the transaction is not currently scheduled, either because it has completed or because it cannot be scheduled, NOT SCHEDULED appears to the right of this field.

## SQL TOT

The total number of SQL calls issued to DB2 by the transaction currently processing. If the Event Collector is not active, this field contains N/A.

## TRANCODE

The transaction code scheduled for this TP name. This is specified in the APPC TP name definition. It can be:

- An implicit APPC transaction.
- The name of a program associated with an explicit APPC transaction (TPI).

## TRANTYPE

Transaction APPC type:

- IMPLICIT       Implicit transaction (transaction code defined in Stage 1 of IMS system generation).
- TPI            CPI-C driven (dynamic transaction not defined in Stage 1 of IMS system generation).

## DAPPC - Output Allocation Direction

BMC SOFTWARE -----			APPC DETAIL			----- PERFORMANCE MGMT		
SERV ==> DAPPC INPUT 07: 25: 55			INTVL=> 3			LOG=> N TGT==> IMSA		
PARM ==> CONVID=03C8F3200000000B			ROW 1 OF 15			SCROLL=> CSR		
EXPAND: APPCL, APPCA								
LU NAME : RXAPPC			(SIDEINFO NAME)			NETID: USBMC		
ACTIVE CONV: 1			LU STATUS:					
TPNAME : DFSSIDE								
QCNT : 6			TOT ALLOCATION REQUESTS:			3 TP STATUS:		
ENQCT: 6			ALLOCATION FAILURES:			0		
			NORMAL DEALLOC REQUESTS:			3		
			ABNORMAL DEALLOC REQUESTS:			0		
			CONFIRMATION COUNTS:			3		
CONVID : 03C8F3200000000B								
DIRECTION: OUTPUT			CONVERSATION TYPE: BASIC					
MODENAME :			SYNC LEVEL: NONE					

### Description:

This display can be selected by EXPANDING from the APPCA display or by selecting DAPPC from the analyzer service list (Primary Option Menu 1). If you select this service from the list and do not specify a conversation ID, the first conversation found is displayed.

The DAPPC display shown is determined by the allocation direction of the conversation. The display for the output allocation direction shows detail information about an outbound allocation to a partner LU/TP from IMS. It shows the LU, TP, and the conversation. The data displayed for output reflects the information passed to APPC from IMS.

If the IMS control region supports APPC but there is no APPC connection, any data available for this conversation and the following message are displayed:

IMS NOT CONNECTED TO APPC

If the IMS control region does not support APPC, the following message is displayed only:

IMS DOES NOT SUPPORT APPC

### Color:

If you have a color monitor:

Red

Highlights the:

- LU STATUS field if the LU is stopped.
- TP STATUS field if the TP is stopped.

### Select Code:

DAPPC

**Parameter:** CONVID=hex

Where hex is a 16-character hexadecimal ID.

**Note:** If this parameter is not specified, the first conversation found is displayed. The conversation could be in an input allocation direction instead of an output allocation direction, depending upon the conversation found.

**Expand:** DAPPC can be EXPANDED by moving the cursor to the following fields in the EXPAND line and pressing ENTER:

APPCL APPC LU status display.

APPCA APACHE activity summary display.

**Scrolling:** The display is scrollable below TPNAME.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b><u>Area 1</u></b>			
LU NAME : RXAPPC	(SIDEINFO NAME)	NETID:	USBMC
ACTIVE CONV:	1	LU STATUS:	

This area shows the LU name and its status. The descriptions are arranged in alphabetical order.

#### ACTIVE CONV

The count of currently active conversations for this LU name in the allocation direction shown by the DIRECTION field.

#### LU NAME

The LU name of the partner LU to IMS for this conversation. Depending on how the conversation is initiated, this is often not an actual LU name when the conversation is initiated by IMS (the value in the ALLOCATION DIRECTION field is OUTPUT); for example:

- The LU name is actually the name of a SIDEINFO entry in APPC. This results from an allocation request specifying only the SIDEINFO name with no other conversation parameters. SIDEINFO NAME is displayed.
- The LU name is DFSLU. This is the default LU name used by IMS when no LU name is supplied.

#### LU STATUS

The status of this LU:

STO This LU name is stopped. The field is highlighted in red.

TRA Trace is on for this LU name.

**Note:** If the LU name is stopped, this status appears as the left-most status.

NETID

The network ID of the LU name

<b>Area 2</b>				
TPNAME : DFSSIDE				
QCNT :	6	TOT ALLOCATION REQUESTS:	3	TP STATUS:
ENQCT:	6	ALLOCATION FAILURES:	0	
		NORMAL DEALLOC REQUESTS:	3	
		ABNORMAL DEALLOC REQUESTS:	0	
		CONFIRMATION COUNTS:	3	

This area shows the TP name. The descriptions are arranged in alphabetical order.

#### ABNORMAL DEALLOC REQUESTS

The count of deallocation requests to APPC for this LUNAME/TPNAME that ended abnormally since the last IMS cold start.

#### ALLOCATION FAILURES

The total number of allocation failures to this LU/TP. It is derived by subtracting NORMAL DEALLOC REQUEST and ABNORMAL DEALLOC REQUEST from TOT ALLOCATION REQUEST. If a conversation is active, 1 is subtracted from the result.

#### CONFIRMATION COUNTS

The count of confirmations received by this TP.

#### ENQCT

Count of requests queued to all TPs for this LUNAME/TPNAME since the last IMS cold start.

#### NORMAL DEALLOC REQUESTS

The count of deallocation requests to APPC for this LUNAME/TPNAME that ended normally since the last IMS cold start.

#### QCNT

Count of requests that are awaiting execution and are currently queued to this LUNAME/TPNAME.

#### TOT ALLOCATION REQUESTS

The total number of allocation requests to APPC for this TP name since the last IMS cold start.

#### TP STATUS

The status of this TP name:

STO This TP name is stopped. If this is the status for the LU, this line is highlighted in red.

TRA Trace is on for this TP name.

**Note:** This field shows a status only when the value in the DIRECTION field is OUTPUT.

## TPNAME

The TP name used in the initiation of the conversation by IMS using APPC services. If the TP name is not provided when the allocation request is made, this field contains DFSSIDE. This could occur, for example, if the LU name is a SIDEINFO entry name.

If this is an input conversation, this TP name is passed to APPC by the partner LU, as shown in the LU NAME field.

If this is an output TP name, this TP name is passed to APPC by IMS except when the TP name is DFSSIDE.

### **Area 3**

```
CONVID   : 03C8F3200000000B      CONVERSATION TYPE: BASIC
DIRECTION: OUTPUT                  SYNC LEVEL: NONE
MODENAME :
```

This area shows conversation-level data. The descriptions are arranged in alphabetical order.

## CONVERSATION TYPE

Conversation type as:

BASIC                Basic conversation.

MAPPED              Mapped conversation.

## CONVID

The conversation ID used in this APPC conversation.

## DIRECTION

The direction of the allocation request as OUTPUT.

This allocation request is directed as output to the LU name from IMS.  
The conversation is allocated by IMS.

## MODENAME

The mode name used in this conversation.

## SYNC LEVEL

Synchronization level:

CONFIRM              Confirmation is required for CPI-C SEND requests.

NO                    No synchronization.

## DLTCH - Latch Detail

BMC SOFTWARE -----				LATCH DETAIL -----				PERFORMANCE MGMT			
SERV ==>	DLTCH	INPUT	13:21:09	INTVL=>	3	LOG=>	N	TGT==>	IMSA		
PARM ==>	MSDB-CUSTDB			SCROLL =>	N/A						
	-- WAIT-COUNT--				-----OWNER-----			----WAITING REGIONS-----			
LATCH ID	CUM	CURR	STAT	RGN	STAT	PSBNAME	RGN1	RGN2	RGN3	RGN4	RGN5
-----											
MSDB-CUSTDB	N/A	0	EXCL	001	ACTV	PSB1MSDB	002				
MSDB-CUSTORDR	N/A	2	EXCL	008	ACTV	PSB2MSDB	007	006			
MSDB-CUSTPART	N/A	0	EXCL	002	ACTV	PSB2MSDB					

**Description:** IMS uses latches to serialize resource access to prevent loss of control at critical times. Use this display to find the source of any generic latch contention.

**Select Code:** DLTCH

**Parameter:** To display detail statistics for generic latches, you can EXPAND from LATCH for a selected latch or enter the following in the parameter line (line 3) of the display:

genltch[-id][,H|W]

where:

genltch Can be any of the following IMS generic latch names:

CBTS  
DBBP  
DMSH  
DMSY  
MSDB

A scrollable list of latches for the requested generic name is displayed.

-id Is a user-defined identifier for a specific latch within the specified IMS generic latch. For example, entering:

CBTS-DPST

in the parameter line puts the CBTS-DPST latch at the top of a scrollable list of CBTS latches. The identifiers that can be specified for each generic latch are:

Latch Example	Description
CBTS-DPST	CBTE name
DBBP-1024	OSAM subpool size
DMSH-AREA1	Fast Path DEDB share area name

DMSY-AREA24      Fast Path DEDB SYNC area name

MSDB-CUSTDB      Fast Path MSDB name

,H      Shows latches that are being held. It can be entered in the parameter line with a generic IMS latch name or with the name and a latch ID. For example:

CBTS, H

displays all the CBTS latches being held.

,W      Is the default. Shows only those latches that are being held and for which resources are waiting. It can be entered in the parameter line with a generic IMS latch name or with the generic latch and a latch ID. For example:

CBTS, W

displays all the CBTS-held latches for which resources are waiting.

**Field Descriptions:**      Each of the fields is shown and described below by display area.

<b>Area 1</b>		
	-- WAIT- COUNT- --	
LATCH ID	CUM	CURR
-----	----	----
MSDB- CUSTDB	N/A	0
MSDB- CUSTORDR	N/A	2
MSDB- CUSTPART	N/A	0

This area lists requested latches and shows how many regions are waiting for each latch. The descriptions are arranged in alphabetical order.

**LATCH ID**

A scrollable list of requested latches.

**--WAIT-COUNT--**

The number of times the resource waited for a latch:

CUM      The cumulative wait count for generic latches is not available.

CURR      The number of regions currently waiting for this latch.

## Area 2

```
-----OWNER-----
STAT  RGN  STAT  PSBNAME
-----
EXCL  001  ACTV  PSB1MSDB
EXCL  008  ACTV  PSB2MSDB
EXCL  002  ACTV  PSB2MSDB
```

This area identifies the latch owner(s). The descriptions are arranged in alphabetical order.

### ---OWNER---

The entries in this column correspond to the STAT values. If STAT is SHR, this column contains the number of regions sharing the latch, such as:

```
-----OWNER-----
RGN  STAT  PSBNAME
-----
SHR- CNT=    1
```

If STAT is EXCL, the OWNER fields identify the exclusive owner of the latch, for example:

```
-----OWNER-----
RGN  STAT  PSBNAME
-----
001  ACTV  CUSTHISM
```

PSBNAME      Either PSB name or INTERNAL (latch is held by an internal IMS function).

RGN            Region ID number of the exclusive owner of this latch. A 000 value indicates that the owner is an IMS internal ITASK.

STAT           Current status can be:

WAIT      Owner is waiting

ACTV      Owner is active

### STAT

The current status of the latch can be:

EXCL           Exclusive

SHR            Shared

**Area 3**

```
-----WAITING REGIONS-----  
RGN1 RGN2 RGN3 RGN4 RGN5  
-----  
002  
007 006
```

This area shows the regions waiting for the requested latches.

---WAITING REGIONS---

The identification number of the region(s) waiting for the latch is shown in the RGNn fields (a maximum of five regions per latch can be displayed). If a region is currently holding another latch, the region ID number is preceded by an \* character.

## DPOOL - Detail Pool (Non-CBT Fixed Pool)

BMC SOFTWARE -----				POOL DETAIL DISPLAY				----- PERFORMANCE MGMT			
SERV ==> DPOOL				INPUT 15: 28: 03		INTVL=> 3		LOG=> N		TGT==> 17AM3BCT	
PARM ==> LUMP, SORT=FS						ROW 1		OF		8 SCROLL=> CSR	
NAME: LUMP				SP: 0		HI - MARK: 103, 336		%HI: 0		EXP+CMP: 4	
TYPE: FIXED				LOC: EPRV		CURR: 66, 224		%CU: 0		EXP - CMP: 2	
RGN: CTL						MAX- SZ: NOLIMIT				WASTED: 1, 960	
BUFFER -BFR/BLOCK-				----- BLOCK COUNT-----				--- BFR COUNT---			
SIZE	PRI	SEC	WA/BFR	CURR	FREE	HI -MRK	EX+CMP	EX -CMP	CURR	HI -MRK	
136	32	32	0	0	0	0	0	0	0	0	0
264	32	16	0	0	0	0	0	0	0	0	0
520	32	16	0	0	0	0	0	0	0	0	0
1, 032	32	16	0	0	0	0	0	0	0	0	0
3, 072	12	12	360	0	0	1	2	0	0	1	1
2, 056	16	8	282	1	0	1	1	1	1	2	2
4, 104	8	8	472	1	0	1	1	1	1	1	1
33, 032	4	2	0	0	0	0	0	0	0	0	0

**Description:** This display shows detail information about a non-CBT fixed pool.

**Select Code:** DPOOL

**Parameter:** Multiple parameters must be separated by a comma.

Pool selection:

id Where id is a valid four-character fixed pool ID.

Valid fixed pool IDs:

- AIOP
- CESS
- CIOP
- EMHB
- FPWP
- HIOP
- LUMP
- LUMC

This DPOOL display can be requested by its Service Select Code with a valid fixed pool ID or by moving the cursor to a fixed pool in the POOLS display and pressing ENTER.

Display sort:

SORT You can enter:

, SORT|S0=cc

where cc can be any of the following two characters. The fixed pool detail display is sorted by BUFFER SIZE by default. The sort sequence is ascending for alphanumeric characters and descending for numeric characters.

**Tip**

An easy way to sort is to move the cursor to the column heading and press ENTER.

The following SORT parameter descriptions are arranged alphabetically. SORT keywords that start with a special character are described first.

BC	Sorts the list by BLOCK COUNT: CURRENT.
BF	Sorts the list by BLOCK COUNT: FREE.
BH	Sorts the list by BLOCK COUNT: HI-MARK.
BS	Sorts the list by BUFFER SIZE (default).
FC	Sorts the list by BFR COUNT: CURRENT.
FH	Sorts the list by BFR COUNT: HI-MARK.
FP	Sorts the list by BFRS PER BLOCK: PRIMARY.
FS	Sorts the list by BFRS PER BLOCK: SECOND.
NE	Sorts the list by EX-CMP (net expansions).
TO	Sorts the list by EX+CMP (total expansions and compressions).
WA	Sorts the list by WA/BFR (average wasted space per buffer request).

**Sorting:**

The display list can be sorted by:

- Using the SORT parameter.
- Positioning the cursor with the TAB key to the heading of the column to be sorted and pressing ENTER.

This overrides any SORT parameter entered in the PARM field and primes the field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by BUFFER SIZE by default. Alphanumeric fields are sorted in ascending order and numeric fields are sorted in descending order.

**Scrolling:**

The list is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>					
NAME: LUMP	SP: 0	HI - MARK: 103,336	%HI: 0	EXP+CMP: 4	
TYPE: FIXED	LOC: EPRV	CURR: 66,224	%CU: 0	EXP - CMP: 2	
RGN: CTL		MAX - SZ: NOLIMIT		WASTED: 1,960	

This area shows the pool identification, location, and statistics. The descriptions are arranged in alphabetical order.

%CU

The current allocation size percentage:

$CURR / MAX - SZ * 100$

%HI

The high-water mark percentage:

$HI - MRK / MAX - SZ * 100$

CURR

The total current pool allocation size.

EXP+CMP

The total number of expansions and compressions of blocks for this pool.

EXP-CMP

The difference between total expansions and compressions (net expansions) of blocks for this pool.

HI-MARK

The high-water mark allocation size since IMS startup.

LOC

The storage location of the pool can be:

CSA

Common storage area

ECSA

Extended CSA

EPRV

Extended private

PRV

Private

MAX-SZ

The upper-size-limit of the fixed pool. An \* character indicates this limit is reached. If there is no limit, NOLIM is displayed.

## NAME

The pool identification can be:

pppp/dddd

or

pppp

where:

dddd           Is a pool descriptor.

pppp           Is the valid IMS pool ID.

Pool IDs are defined in the IBM *IMS System Definition Reference* manual. The descriptor added to the pool ID in the POOLS display is a more common term than the pool ID.

### AOIP

A fixed automated operator interface buffer pool.

### CESS

A fixed external subsystem work pool.

### CIOP

A fixed pool for terminal buffers.

### EMHB

A fixed pool for IFP message-driven regions.

### FPWP

A Fast Path fixed work pool.

### HIOP

A fixed communication I/O buffer pool.

### LUMC

A fixed pool for the APPC/IMS LU6.2 function.

### LUMP

A fixed pool for the APPC/IMS LU6.2 function.

## RGN

Pool ownership by region:

CTL           Control region.

DLS           DLI address space.

## SP

The OS/390 subpool number where this pool is allocated.

## TYPE

The type of pool as:

### FIXED

Up to 32 different buffer sizes can be specified for a fixed pool. For each buffer size, you can specify the buffer count for the primary block and the buffer count for the secondary blocks. There is only one primary block per buffer size for a given fixed pool. When the space in the primary block is exhausted, a secondary block is allocated (expansion). Subsequent secondary blocks are allocated (expansion) as needed. You can specify an upper-limit size for a fixed pool.

#### Tuning Tip

- Secondary blocks (and potentially the primary block if specified) can be deleted (compressed) when they are no longer used. Expansions (allocation of secondary blocks) should occur only for a heavy workload. Performance would be degraded if new blocks are frequently being allocated (expansion) and deleted (compression). The primary blocks should handle most of the normal workload, while the secondary blocks handle the heavy workload.

If specific buffer sizes are rarely used (for example, once a week for special processing), you can specify the primary blocks for those buffer sizes as compressible. This way, the primary blocks are allocated only when they are needed, as opposed to being allocated when the pool is initialized. Those blocks are deleted (compressed) when they are not needed. The DPOOL service can be used to determine the usage of all buffer sizes for a given fixed pool.

- Since the buffers in a fixed pool are fixed sizes (there are 32 maximum buffer sizes allowed), some of the spaces in the buffers would be wasted. A small amount of wasted space is unavoidable. However, a large amount of wasted storage should be avoided because it can impact performance. If the value is large, use the DPOOL service to determine which buffer sizes can be adjusted (if any). If a specific buffer size uses only a few buffers (for example, 1), the primary buffer count per block can be reduced.

## WASTED

The accumulative wasted space for all buffer sizes in this pool since IMS started.

<b>Area 2</b>											
BUFFER SIZE	-BFR/BLOCK-		---- BLOCK COUNT----							--- BFR COUNT---	
	PRI	SEC	WA/BFR	CURR	FREE	HI - MRK	EX+CMP	EX- CMP		CURR	HI - MRK
136	32	32	0	0	0	0	0	0		0	0
264	32	16	0	0	0	0	0	0		0	0
520	32	16	0	0	0	0	0	0		0	0
1,032	32	16	0	0	0	0	0	0		0	0
3,072	12	12	360	0	0	1	2	0		0	1
2,056	16	8	282	1	0	1	1	1		1	2
4,104	8	8	472	1	0	1	1	1		1	1
33,032	4	2	0	0	0	0	0	0		0	0

This area shows detailed pool data by buffer size. The descriptions are arranged in alphabetical order.

**-BFR/BLOCK-**

**PRI**            The number of user-defined buffers in the primary block. There is only one primary block for a given buffer size.

**SEC**            The number of user-defined buffers in the secondary blocks.

**---BFR COUNT---**

**CURR**  
The number of currently used buffers.

**HI-MRK**  
The maximum buffer counts since initialization.

**---BLOCK COUNT---**

**CURR**            The number of currently allocated blocks. The secondary blocks can be compressed.

**FREE**            The number of free blocks not compressed.

**HI-MRK**          The maximum block counts since initialization.

**BUFFER SIZE**

The defined buffer size. A maximum of 32 different buffer sizes can be defined in a fixed pool. An asterisk (\*) indicates this is an overflow buffer size.

**EX+CMP**

The total number of expansions and compressions of blocks for this buffer size.

**EX-CMP**

The difference between the expansions and compressions counts (net expansions) of blocks for this buffer size.

**WA/BFR**

Average wasted space per buffer request for this buffer size.

**Tuning Tips**

Since the buffers in a fixed pool are of fixed sizes (there are 32 maximum buffer sizes allowed), some of the spaces in the buffers would be wasted. A small amount of wasted space is unavoidable. However, a large amount of wasted storage should be avoided because it can impact performance. If the value is large, use the DPOOL service to determine which buffer sizes can be adjusted (if any). Moreover, if a certain buffer size uses only a few buffers (for example, 1), the primary buffer count per block can be reduced.

## DPOOL - Detail Pool (Non-CBT Variable Pool)

BMC SOFTWARE -----		POOL DETAIL DISPLAY		----- PERFORMANCE MGMT	
SERV ==> DPOOL		INPUT 15:37:09	INTVL=> 3	LOG=> N	TGT==> I7AMBCT
PARM ==> PSBW					SCROLL=> CSR
NAME: PSBW	SP: 231	HI - MRK:	5,720	%HI:	23
TYPE: VARIABLE	LOC: ECSA	CURR:	5,720	%CU:	23
RGN: DLS		FREE:	18,856	MAX-SZ:	24,576
	BLOCK CNTS	SMALLEST	AVERAGE	LARGEST	
FREE:	1	18,856	18,856	18,856	
ALLOCATED:	1	5,720	5,720	5,720	

**Description:** This display shows detail information for the selected non-CBT variable pool.

**Select Code:** DPOOL

**Parameter:** A valid four-character variable pool ID.

Valid variable pool IDs:

- DBWP
- DLDP
- DLMP
- DPSB
- EPCB
- MAIN
- MFBP
- PSBW (default)
- QBUF

This DPOOL display can be requested by its Service Select Code with a valid variable pool ID or by moving the cursor to a variable pool in the POOLS display and pressing ENTER.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>					
NAME: PSBW	SP: 231	HI - MRK:	5,720	%HI:	23
TYPE: VARIABLE	LOC: ECSA	CURR:	5,720	%CU:	23
RGN: DLS		FREE:	18,856	MAX-SZ:	24,576

This area shows the pool identification, location, and statistics. The descriptions are arranged in alphabetical order.

%CU

The current size allocation percentage:

$$\text{CURR} / \text{MAX-SZ} * 100$$

This value is 0 when the displayed pool has no limit.

%HI

The high-water mark percentage:

$$\text{HI-MRK} / \text{MAX-SZ} * 100$$

This value is 0 when the displayed pool has no limit.

CURR

The current total pool allocation size.

FREE

The total length of all free spaces in the variable pool.

HI-MRK

The high-water mark pool allocation size since IMS startup.

LOC

The storage location of the pool can be:

CSA

Common storage area

ECSA

Extended CSA

EPRV

Extended private

PRV

Private

MAX-SZ

The total pool size in bytes.

NAME

The pool identification can be:

pppp/dddd

or

pppp

where:

dddd      Is a pool descriptor.

pppp      Is the valid IMS pool ID.

Pool IDs are defined in the IBM *IMS System Definition Reference* manual. The descriptor added to the pool ID in the DPOOL display is a more common term than the pool ID.

**DBWP/DMBW**

The DBWP (DMBW) pool is the pool for the DMB work areas.

**DLDP/DMBP**

The DLDP (DMBP) pool holds DMBs.

**DLMP/PSBC**

The DLMP (PSBC) pool is the total PSB pool in the:

- Control region (CTL) when the IMS option LSO is not S.
- CSA partition of the PSB pool when LSO=S.

**DPSB/PSBD**

The DPSB pool is the DL/I SAS private partition of the PSB pool. This appears only when LSO=S.

**EPCB**

This pool is required for Fast Path database access.

**MAIN/WKAP**

MAIN (WKAP) is the general work pool.

**MFBP**

MFS pool.

**PSBW**

PSB work pool.

**QBUF**

QMGR buffer pool.

**RGN**

Pool ownership by region:

CTL      Control region.

DLS      DLI address space.

**SP**

The OS/390 subpool number where this pool is allocated.

**TYPE**

The type of pool as:

**VARIABLE**

**Area 2**

	<b>BLOCK CNTS</b>	<b>SMALLEST</b>	<b>AVERAGE</b>	<b>LARGEST</b>
<b>FREE:</b>	<b>1</b>	<b>18, 856</b>	<b>18, 856</b>	<b>18, 856</b>
<b>ALLOCATED:</b>	<b>1</b>	<b>5, 720</b>	<b>5, 720</b>	<b>5, 720</b>

This area shows fragmentation information. The descriptions are arranged in alphabetical order.

**ALLOCATED: AVERAGE**

The average length of the allocated blocks in the pool.

**ALLOCATED: BLOCK CNT**

The number of allocated blocks in the pool.

**ALLOCATED: LARGEST**

The length of the largest allocated block in the pool.

**ALLOCATED: SMALLEST**

The length of the smallest allocated block in the pool.

**FREE: AVERAGE**

The average length of the free spaces in the pool.

**FREE: BLOCK CNT**

The number of free block counts in the pool.

**FREE: LARGEST**

The length of the largest free space in the pool.

**FREE: SMALLEST**

The length of the smallest free spaces in the pool.

## LATCH - Latch Summary

BMC SOFTWARE ----- LATCH SUMMARY----- PERFORMANCE MGMT													
SERV ==> LATCH INPUT		11: 44: 25 INTVL=> 3 LOG=> N TGT==> IMSxxx											
PARM ==> SORT=LA		LINE 1 OF 50 SCROLL==> CSR											
EXPAND: MON(INTNL) , LINESEL(DLTCH)													
CURR													
LATCH NAME	ID	WAIT CNT	ST	RGN	STAT	PSBNAME	RGN1	RGN2	RGN3	RGN4	WAIT TOT (s)	CNT	AVG (us)
ALLOC PSB	APSB	2	EX	003	WAIT	PHDAMI NQ	*002	004			0	1564	584
BLK LOADER	DBLR	0	EX	001	ACTV	CUSTHI SM					1	23K	44
CBTS LQB	VLQB	0				-----GENERIC-----					2	44K	423
*CBTS POOL	CBTS	0				-----GENERIC-----					23	433K	52
CBTS VTCB	VTCB	0				-----GENERIC-----					0	245	874
CCTL	CCTL	0									0	34	643
CONV CHKPT	CONV	0									0	346	34
DB CHKPT	DBSL	0									0	557	878
DBRC AUTH	DBAU	0									0	574	537
DC CHKPT	DCSL	0									3	76K	45
DDIR BLK	DDRB	0				-----GENERIC-----					0	579	246
DDIR POOL	DDRP	0									46	47K	970
DEP REGION	DBLK	0									5	643	7656
DISPATCHER	DISP	0									0	4334	34
*DMAC SHARE	DMSH	N/A				-----GENERIC-----					N/A	N/A	N/A
*DMAC SYNC	DMSY	N/A				-----GENERIC-----					N/A	N/A	N/A
DMB BLK	DMBB	0				-----GENERIC-----					0	234	975
DMB POOL	DMBP	0	EX	006	WAIT	PHDARHDR					0	423	48
FP-CMD	N/A	0									N/A	N/A	N/A
FP-RESOURCE	N/A	0									N/A	N/A	N/A
FP-SYNC	N/A	0	EX	005	ACTV	DBFCST00					N/A	N/A	N/A
LOG	LOGL	0	EX	000	ACTV	INTERNAL					0	423	54
APPC INPUT	LUBT	N/A				-----GENERIC-----					N/A	N/A	N/A
LU 6.2 LUM	LUML	0									0	0	0
*MSDB	MSDB	N/A				-----GENERIC-----					N/A	N/A	N/A
*OSAM BUFP	DBBP	0				-----GENERIC-----					0	6756	57
PDIR BLK	PDRB	0				-----GENERIC-----					0	24K	12
PDIR POOL	PDRP	0									0	68	867
PI ENQ, DEQ	XCNQ	0									6	8547	687
PSB BLK	PSBB	0				-----GENERIC-----					0	43	24
PSB POOL	PSBP	0									3	423	6898
QUEUE BUFFER	QBSL	0				-----GENERIC-----					0	35	245
QUEUE MGR	QMGR	0				-----GENERIC-----					41	53K	765
SCHED SUBQ	SUBQ	0				-----GENERIC-----					0	76	757
SCHEDULE	SCHD	0									2	43K	35
SMB HASH	BLKM	0				-----GENERIC-----					0	35	345
STATISTICS	ACTL	0									2	42K	58
STORAGE MGR	SMGT	0				-----GENERIC-----					0	357	976
TCT BLOCK	TCTB	0				-----GENERIC-----					0	658	234
TERMINAL	TERM	0									0	243	5
USER	USER	0									0	3463	66
***** END OF DATA *****													

### Description:

LATCH shows detailed contention information for nongeneric latches and summary statistics for generic latches. To see contention statistics for selected generic latches, you can use the DLTCH service (see “Expand” in this service description).

IMS uses latches to serialize task access to resources to prevent loss of control at critical times. Use this LATCH service display to find the source of any latch contention. The display shows statistics for the latches listed in the first column. These latches may be used by multiple regions. If more than one region is waiting for a latch, the regions waiting for the latches are shown by their identification number in the RGNn fields (RGN1 through RGN4).

**Select Code:** LATCH

**Parameter:** You can enter:

`SORT|S0=cc`

where cc can be any of the following two characters. The display list is sorted by CURR WAIT CNT and LATCH NAME, respectively, by default.

AT      Sorts the list by AVG WAIT TIME.

CC      Sorts the list by CURR WAIT CNT and LATCH NAME, respectively (default).

ID      Sorts the list by ID.

LA      Sorts the list by LATCH NAME.

TC      Sorts the list by TOT WAIT CNT.

TT      Sorts the list by TOT WAIT TIME.

**Expand:** This LATCH display can be EXPANDED by moving the cursor to the following fields and pressing ENTER:

MON(INTNL)      Active Timer Requests display of all active IMS internals monitors (INTNL area).

DLTCH              Detail latch display.

Move the cursor to one of the generic latches with an \* character in front of its LATCH NAME and press ENTER to expand that latch in the DLTCH display.

**Sorting:** The display list can be sorted by:

- Using the SORT parameter.
- Positioning the cursor with the TAB key to the column heading to be sorted and pressing ENTER.

This overrides any SORT parameter entered in the PARM field and primes the field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by CURR WAIT CNT as the primary sort default and by LATCH NAME as the secondary sort. Alphanumeric fields are sorted in ascending order and numeric fields are sorted in descending order.

**Scrolling:** The display list is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<u>Area 1</u>			
LATCH NAME		ID	CURR WAIT CNT
-----		----	----
ALLOC PSB	APSB		2
BLK LOADER	DBLR		0

This area lists the latches and shows how many regions are waiting for each latch. The descriptions are arranged in alphabetical order.

#### CURR WAIT CNT

Number of regions currently waiting for the latch.

#### ID

The IMS latch ID associated with the descriptive latch name (see “LATCH NAME” below).

#### LATCH NAME

The first column is a descriptive latch name for the IMS latch ID shown in the second column. A generic latch that can be expanded has an \* character in front of its name. The latches that are displayed include:

ALLOC PSB	IMS latch ID: APSB
	A block mover latch that serializes PSB allocation processing.
APPC INPUT	IMS latch ID: LUBT
	It serializes the IMS LU6.2 blocks (TIBs) used to initiate an LU6.2 conversation.
BLK LOADER	IMS latch ID: DBLR
	This latch serializes the loading of a PSB when it is not found in memory.
CBTS LQB	IMS latch ID: VLQB
	It serializes access of CBTS LQB control blocks.
CBTS POOL	IMS latch ID: CBTS
	It serializes access of CBTS pool control blocks.
CBTS VTCB	IMS latch ID: VTCB
	It serializes access of CBTS VTCB control blocks.

CCTL	IMS latch ID: CCTL It serializes the access of a DBCTL resource.
CONV CHKPT	IMS latch ID: CONV The conversational latch serializes checkpoint and conversation processing.
DB CHKPT	IMS latch ID: DBSL It serializes DB checkpoint processing because some control blocks cannot be changed in the middle of a checkpoint.
DBRC AUTH	IMS latch ID: DBAU A block mover latch that serializes the move of DBRC authorization blocks.
DC CHKPT	IMS latch ID: DCSL It serializes DC checkpoint processing because some control blocks cannot be changed in the middle of checkpoint.
DDIR BLK	IMS latch ID: DDRB A block mover latch that serializes the move of DDIR blocks.
DDIR POOL	IMS latch ID: DDRP A block mover latch that serializes the access of DDIR pool space required for a DDIR.
DEP REGION	IMS latch ID: DBLK It serializes access of dependent region control blocks.
DISPATCHER	IMS latch ID: DISP It serializes the usage of some of the IMS system dispatcher's control blocks.
DMAC SHARE	IMS latch ID: DMSH These latches control Data Entry DataBase (DEDB) access. There is one latch per DEDB.
DMAC SYNC	IMS latch ID: DMSY These latches control DEDB synchronization. There is one latch per DEDB.
DMB BLK	IMS latch ID: DMBB A block mover latch that serializes the move of a DMB block.
DMB POOL	IMS latch ID: DMBP A block mover latch that serializes the access of DMB pool space required for a DMB block move.

The following, FP-CMD, FP-RESOURCE, and FP-SYNC, are designated as a pseudo-latch, because they are not managed by DFSCLM00.

FP-CMD	IMS latch ID: Not applicable  The Fast Path command pseudo-latch serializes any IMS commands relating to Fast Path with Fast Path processing.
FP-RESOURCE	IMS latch ID: Not applicable  The exclusive control Fast Path resource pseudo-latch (IMS XCGL) serializes Fast Path buffers.
FP-SYNC	IMS latch ID: Not applicable  Fast Path checkpoint synchronization pseudo-latch controls the execution of synchronization process modules for Fast Path regions.
LOG	IMS latch ID: LOGL  It serializes IMS log processing.
LU 6.2 LUM	IMS latch ID: LUML  It serializes IMS LU6.2 control blocks.
MSDB	IMS latch ID: MSDB  These latches control the locks on the Main Storage DataBase (MSDB). There is one latch per MSDB.
OSAM BUFP	IMS latch ID: DBBP latch  It serializes OSAM database buffer pools. There is one latch for each subpool.
PDIR BLK	IMS latch ID: PDRB  A block mover latch that serializes the move of PDIR blocks.
PDIR POOL	IMS latch ID: PDRP  A block mover latch that serializes the access of PDIR pool space required for a PDIR block move.
PI ENQ,DEQ	IMS latch ID: XCNQ  It serializes enqueue/dequeue for program isolation.
PSB BLK	IMS latch ID: PSBB  A block mover latch that serializes the move of a PSB block.
PSB POOL	IMS latch ID: PSBP  A block mover latch that serializes the access of PSB pool space required for a PSB block move.
QUEUE BUFFER	IMS latch ID: QBSL  It serializes access of queue buffers.
QUEUE MGR	IMS latch ID: QMGR  It serializes storage management.

SCHED SUBQ	IMS latch ID: SUBQ It serializes access of scheduler subqueues. There is one latch for each scheduler subqueue.
SCHEDULE	IMS latch ID: SCHD It protects the scheduling process from online change and allows synchronization with other IMS modules involved in the scheduling process.
SMB HASH	IMS latch ID: BLKM It serializes access of SMB hash tables that are required for dynamic SMB (transactions) as a result of CPI-C transactions.
STATISTICS	IMS latch ID: ACTL It serializes statistics control blocks.
STORAGE MGR	IMS latch ID: SMGT It serializes access of storage pool control blocks.
TCT BLOCK	IMS latch ID: TCTB This latch is required to coordinate the activity of SMB chains from TCT (transaction class table). There is one latch for each TCT.
TERMINAL	IMS latch ID: TERM It serializes terminal control block access.
USER	IMS latch ID: USER It serializes the access of user/SPQB blocks.

#### Area 2

```

----- OWNER -----
ST  RGN  STAT  PSBNAME
-----
EX 003 WAIT  PHDAMI NQ
EX 001 ACTV  CUSTHI SM

```

This area identifies the latch owner(s). The descriptions are arranged in alphabetical order.

#### ---OWNER---

The entries in this column correspond to the ST values. If ST is SH, this column contains the number of regions sharing the latch, for example:

```

----- OWNER -----
RGN  STAT  PSBNAME
-----
SHR- CNT=    1

```

If ST is EX, the OWNER fields identify the exclusive owner of the latch as follows:

```

-----OWNER-----
RGN  STAT  PSBNAME
-----
001  ACTV  CUSTHISM

```

PSBNAME      Either PSB name or INTERNAL (latch is held by an internal IMS function).

RGN            Region ID number of the exclusive owner of this latch. A 000 value indicates that the owner is an IMS internal ITASK.

STAT          Current status can either be WAIT (owner is waiting) or ACTV (owner is active).

ST

The current status of the latch as:

EX            Exclusive

SH            Shared

### Area 3

```

----- WAIT -----
- - WAITING REGIONS- - TOT  CNT  AVG
RGN1 RGN2 RGN3 RGN4  (s)      (us)
-----
*002  004                      0 1564  584
                               1  23K   44

```

This area shows the regions waiting for the latch and the amount of wait time. The descriptions are arranged in alphabetical order.

---- WAIT ----

The latch wait time as:

AVG (us)      The average wait time for the latch type in microseconds.

CNT            The total accumulative wait count for the latch type.

TOT (s)        The total wait time for the latch type in seconds. These statistics reflect only the activity since the last IMS checkpoint.

-WAITING REGIONS-

The identification number of the region(s) waiting for the latch is shown in the RGNn fields (a maximum of four regions per latch can be displayed). If a region is currently holding another latch, the region ID number is preceded by an \* character.

## LOGST - Log Statistics

BMC SOFTWARE				LOG STATISTICS				PERFORMANCE MGMT					
SERV ==> LOGST				INPUT		13: 21: 09		INTVL=> 3		LOG=> N		TGT==> IMSxxx	
PARM ==>												SCROLL => N/A	
LOG DATA SET CONFIGURATION													
DDNAME	TYPE	BLKSIZE	DEVICE	VOLUME	DEVN	BLOCK NO	MODE	-- COMMENTS--					
DFSOLP02	P-OLDS	12, 288	3350	BABP03	0250	00000083	SINGLE	73% USED					
DFSWADS0	P-WADS	2, 080	3350	BABP03	0250	00000084	SINGLE						
OLDS STATISTICS													
MAX BLKS THIS OLDS				15		ALLOCATED OLDS				3			
FIRST BLOCK(HEX)				00000079		ARCHIVE COUNT				0			
CURRENT BLOCK(HEX)				00000083		AUTO ARCHIVE				1			
LAST BLOCK(HEX)				00000087		BUFFERS				2			
RECOVERY BLOCK(HEX)				00000076		BUFFER SIZE				12, 288			
AWE WRITE REQUESTS				0		BUFFER WAITS				10			
CHECK WRITE REQS				1, 797		BLOCKS WRITTEN				131			
WAIT WRITE REQUESTS				0		BLOCKS READ				0			
WADS STATISTICS													
NUMBER EXCPS		1, 805		SEGS WRITTEN		2, 320		SEGS PER I/O		1. 29			
CHECKPOINT DATA													
LATEST:		86120/105021		00000076		CHKPT RECORDS				495			
SNAPQ/DUMPQ:		86120/103511		00000002		CHKPT FREQUENCY				5, 000			
OLDEST LCRE:		86120/105021		00000076									

**Description:** LOGST displays checkpoint, configuration, and statistical information on the Online Log Data Set (OLDS) and Write Ahead Data Set (WADS).

**Select Code:** LOGST

**Parameter:** None.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>									
LOG DATA SET CONFIGURATION									
DDNAME	TYPE	BLKSIZE	DEVICE	VOLUME	DEVN	BLOCK NO	MODE	-- COMMENTS--	
DFSOLP02	P-OLDS	12, 288	3350	BABP03	0250	00000083	SINGLE	73% USED	
DFSWADS0	P-WADS	2, 080	3350	BABP03	0250	00000084	SINGLE		

This area shows the configuration of the current log data sets. The descriptions are arranged in alphabetical order.

**BLKSIZE** The block size of the data set. This is always 2080 for the WADS; 2048 segment size plus a 32-byte suffix. The OLDS block size is obtained from the DSCB for the data set and must be a multiple of 2048. All OLDSs must have the same block size.

**BLOCK NO** The last sequential block number (hexadecimal) written to the data set. The WADS block number is ahead of or the same as the OLDS block number.

COMMENTS Could be one of the following:

- xx% USED

where xx is the percentage of blocks that are used in the current OLDS. The WADS does not show the percentage because it is a preformatted data set that is continually reused.

- DCB NOT OPEN

This message appears in the comments field whenever LOGST detects that the data set is not open. This occurs when IMS is in the process of switching to a new OLDS.

- WADS NOT ACTIVE

This message appears in the comments field whenever the WADS is not active, which can occur when all available WADS and spares have I/O errors. This causes degradation because IMS must write directly to the OLDS and pad and truncate blocks when CHECK WRITES are requested.

DDNAME The current ddname being used by IMS.

DEVICE The type of DASD device where the data set resides. If ?DEVT? appears, it indicates an unknown DASD device type. If ?UNIT? appears, it indicates an unsupported device type. BMC Software should be contacted to include these device types.

DEVN The address of the volume containing the data set.

MODE Indicates whether SINGLE or DUAL OLDS/WADS was specified at IMS initialization.

TYPE Identifies the type of data set. The first character is either a P for primary or S for secondary. The character P or S is followed either by OLDS for the Online Log Data Set or by WADS for the Write Ahead Data Set.

VOLUME The name of the volume containing the data set.

<u>Area 2</u>			
----- OLDS STATISTICS -----			
MAX BLKS THIS OLDS	15	ALLOCATED OLDS	3
FIRST BLOCK(HEX)	00000079	ARCHIVE COUNT	0
CURRENT BLOCK(HEX)	00000083	AUTO ARCHIVE	1
LAST BLOCK(HEX)	00000087	BUFFERS	2
RECOVERY BLOCK(HEX)	00000076	BUFFER SIZE	12,288
AWE WRITE REQUESTS	0	BUFFER WAITS	10
CHECK WRITE REQS	1,797	BLOCKS WRITTEN	131
WAIT WRITE REQUESTS	0	BLOCKS READ	0

This area shows OLDS status and logger activity. The descriptions are arranged in alphabetical order.

#### ALLOCATED OLDS

Number of OLDSs that IMS has allocated.

#### ARCHIVE COUNT

This figure is used to determine when IMS needs to issue an archive request to DBRC if automatic archiving is selected (see AUTO ARCHIVE below). It reflects the number of OLDS filled since the last archive request by IMS. ARCHIVE COUNT never exceeds AUTO ARCHIVE and does not reflect the OLDS that still require archiving.

#### AUTO ARCHIVE

User specified number of OLDSs to be filled before IMS schedules an archive job. Zero means no automatic archive (see ARCHIVE COUNT above). When the ARCHIVE COUNT reaches the AUTO ARCHIVE limit, IMS issues an archive request to DBRC.

#### AWE WRITE REQUESTS

Number of asynchronous write requests to the IMS logger under an Asynchronous Work Element (AWE). IMS Fast Path is the prime user of this type of request. It issues an AWE write request to chain together multiple log records to be written in a single request to the logger.

#### BLOCKS READ

Number of blocks read from the OLDSs since IMS restart. Blocks are read only for backout processing.

#### BLOCKS WRITTEN

Number of physical blocks written to the OLDS since IMS restart.

#### BUFFER SIZE

The buffer size of the OLDSs.

#### BUFFER WAITS

Number of times the logger had to wait for a buffer. An attempt was made to place a logical record in a physical log buffer, but the buffers were not written to the OLDS and no buffer was available.

#### BUFFERS

Number of access method buffers used for the OLDS. This number is specified on the DD statement for the OLDS by the DCB=BUFNO= keyword. There are no separate WADS buffers because WADS uses the same buffers as OLDS, but each buffer is segmented into 2K segments.

#### CHECK WRITE REQ

Number of check write requests to the logger. This type of request forces a physical write to the WADS/OLDS, if not already done. The requestor is suspended until the write is done. The main use of this type of request is for log write ahead for database updates.

#### CURRENT BLOCK(HEX)

Hexadecimal sequence number of the last block physically written to the OLDS.

#### FIRST BLOCK(HEX)

Hexadecimal sequence number of the first block on the current OLDS.

LAST BLOCK(HEX)

Hexadecimal sequence number of the last block that could be written to the current OLDS.

MAX BLKS THIS OLDS

Total capacity, in blocks, of the current OLDS.

RECOVERY BLOCK(HEX)

Hexadecimal sequence number of the most recent block needed for recovery in the event of a restart. This number is obtained from the oldest local recovery element (LCRE).

**Tip**

If this number is far behind the CURRENT number, it is most likely a BMP that is executing and not taking sufficient checkpoints.

WAIT WRITE REQUESTS\

Number of wait write requests to the logger. This type of request is the same as a check write request with the exception that the request does not have to be immediately satisfied. The requestor is prepared to wait until the block is written, which happens after someone else issues a CHKW request; the buffer then fills up, or 1 second elapses on an internal timer. The request could be satisfied immediately or the requestor could wait a maximum of 1 second. The prime user of this type of request is the DC Log Write Ahead (DCLWA) option, which ensures that a terminal output message is placed on the log before it is sent to the terminal.

**Note:** IMS does not keep statistics for ordinary write requests to the log; therefore, the total of the above statistics for the logger is not a total of all requests.

**Area 3**

----- WADS STATISTICS -----			
NUMBER EXCPS	1, 805	SEGS WRITTEN	2, 320
		SEGS PER I/O	1. 29

This area displays statistics about WADS activity. The descriptions are arranged in alphabetical order.

NUMBER EXCPS

Number of physical I/O requests issued to each WADS. If dual WADS logging is selected, the actual number of EXCPs is twice this amount.

SEGS PER I/O

Average number of segments written per I/O operation. IMS attempts to chain as many segments together as possible to reduce the number of physical I/Os.

SEGS WRITTEN

Number of segments written to the WADS. IMS segments each log buffer into 2K segments.

**Area 4**

```
----- CHECKPOINT DATA -----
LATEST:      86120/105021 00000076   CHKPT RECORDS           495
SNAPQ/DUMPQ: 86120/103511 00000002   CHKPT FREQUENCY        5,000
OLDEST LCRE: 86120/105021 00000076
```

This area displays the IMS checkpoint data. The descriptions are arranged in alphabetical order.

**CHKPT FREQUENCY**

Number of logical log records between IMS checkpoints, specified in the IMS system definition.

**CHKPT RECORDS**

Current count of logical log records written to the log since the last checkpoint. This count is reset when it reaches CHKPT FREQUENCY and a checkpoint is issued.

**LATEST**

Latest IMS checkpoint identifier and hexadecimal log block sequence number.

**OLDEST LCRE**

Oldest IMS checkpoint identifier and hexadecimal log block sequence required to back out a currently running program or transaction. LCRE is an IMS internal control block used for recovery purposes.

**SNAPQ/DUMPQ**

The IMS checkpoint identifier and hexadecimal log block sequence number when the last /CHE SNAPQ or DUMPQ was issued. If a SNAPQ or DUMPQ is not issued, this is the last cold start checkpoint.

## PI - Program Isolation

BMC SOFTWARE	-----	PROGRAM ISOLATION	-----	PERFORMANCE MGMT
SERV ==> PI		INPUT 13: 21: 09	INTVL=> 3	LOG=> N
PARM ==>			LINE 1 OF	16
				SCROLL => CSR
				<< EXPAND >>
-----				
	PI	POOL SPACE SUMMARY	-----	
		DYNAMIC POOL		INTERNAL POOL
		BYTES	QCBS	BYTES QCBS
CURRENT POOL SIZE		0	0	1, 584 66
INCREMENT		1, 024	42	
MAXIMUM POOL SIZE		4, 096	170	
FREE SPACE		0	0	1, 584 66
-----				
	PI	ENQUEUE SUMMARY	-----	
RGN	PROGRAM	TRANSACT	ENQCNTN	READS UPDATES EXCL
1	PIWAIT1	PIWAIT1	1	1 0 0
2	PIWAIT2	PIWAIT2	2	0 2 0
				1 CUSTHISM 1 0000165C

**Description:** The PI service allows investigation into program isolation problems. All dependent regions with outstanding enqueues are shown. Any region holding a resource causing a wait is highlighted. For regions in a PI wait, information about the lock holder is displayed to the right.

**Select Code:** PI

**Parameter:** First region number to display (optional)

**Expand:** The details of the DL/I call being performed by the region can be viewed on the DLIST display. This service can be accessed from the PI display by moving the cursor to the row showing that region and pressing the ENTER key. DLIST shows the call details when the ENTER key is pressed. It is not synchronized with the PI ENQUEUE SUMMARY area of the display. When returning from a DLIST EXPAND, the data shown is the same as when DLIST was selected.

**Scrolling:** The display is scrollable so you can view all the regions without refreshing the data. Data is refreshed when the ENTER key is pressed and no cursor selection is used.

**Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>				
-----				
	PI	POOL SPACE SUMMARY	-----	
		DYNAMIC POOL		INTERNAL POOL
		BYTES	QCBS	BYTES QCBS
CURRENT POOL SIZE		0	0	1, 584 66
INCREMENT		1, 024	42	
MAXIMUM POOL SIZE		4, 096	170	
FREE SPACE		0	0	1, 584 66

This area summarizes the state of the program isolation pools. The descriptions are arranged in alphabetical order.

CURRENT POOL SIZE

The current size of the dynamic pool and internal pool, given both in bytes and the corresponding number of 16-byte queue control blocks (QCBs) that it can contain.

FREE SPACE

The amount of free space in the dynamic pool or internal pool, given both in bytes and the corresponding number of 16-byte QCBs that it can contain.

INCREMENT

The amount the PI pool is expanded in similar units if additional space is required.

MAXIMUM POOL SIZE

The maximum possible size of the dynamic pool, given both in bytes and the corresponding number of 16-byte QCBs that it can contain.

<b>Area 2</b>						
- - - - - PI ENQUEUE SUMMARY - - - - -						
RGN	PROGRAM	TRANSACT	ENQCNT	READS	UPDATES	EXCL
1	PIWAIT1	PIWAIT1	1	1	0	0
2	PIWAIT2	PIWAIT2	2	0	2	0

This area gives a one-line program isolation summary for the regions with outstanding enqueues. If a region holds a resource resulting in a wait of another region, the holding region is highlighted. The resource may be identified by examining the data shown for the waiting region(s) in Area 3 of the display.

The descriptions are arranged in alphabetical order.

ENQCNT

The number of enqueues currently held by the region.

PROGRAM/TRANSACT

The program and transaction, if scheduled.

READS/UPDATES/EXCL

Enqueues, broken down by level.

RGN

The region ID.

<b>Area 3</b>			
- - - - - PI ENQUEUE SUMMARY - - - - -			
	RGN	DB/AREA	DCB RBA
	1	CUSTHISM	1 0000165C

Data appears in this area only for regions in a PI wait. The descriptions are arranged in alphabetical order.

#### DB/AREA

The needed resource is identified by database or area name. If ?????? appears in this field, the DCB field is blank.

#### DCB

The needed resource is identified by DCB number. If ?????? appears, the enqueue is on an internal IMS system resource. An example is the enqueue issued during data set open/close.

#### RBA

The hexadecimal ID used by the Program Isolation routines in IMS to identify the resource. It is most commonly the relative byte address (RBA) of the VSAM control interval (CI) or the relative block number (RBN) of the OSAM block containing the segment in contention. The rules that IMS uses to construct the resource ID are listed below to assist in problem determination.

#### FP DEDB

The ID is the high order three bytes of the RBA of the VSAM control interval containing the segment.

#### FP MSDB

The ID is the address of the segment prefix in the IMS Control Region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

#### HDAM OSAM

The ID is the RBA of the block containing the segment.

#### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (x'40') of the high order byte is turned on.

#### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

#### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

In some cases, Program Isolation uses special resource ID values that do not represent a relative byte address (RBA) In these cases, the following text is displayed in the RBA field:

#### AREA

Used by Fast Path to lock an entire area. This is mostly used during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

DATASET

Used at data set open time for buffer invalidation.

DS-BUSY

Data set busy is used to ensure that no CI or CA split occurs during a read.

EXTEND

Used to serialize data set extension.

NON-DBRC

Database is not registered with DBRC.

RGN     The ID of the region currently holding the needed resource.

## POOLC - Pool Summary (CBT)

BMC SOFTWARE -----					POOL SUMMARY -----					PERFORMANCE MGMT				
SERV ==> POOLC					INPUT	13: 21: 09	INTVL=> 3		LOG=> N	TGT==> IMSxxx				
PARM ==> AWE										SCROLL => N/A				
ID	RGN	LOC	FX	SP	SIZE	HWM	BLK	BLKTH	GETM	FREEM	COMMENTS			
I OSB	CTL	CSA	N	228	8, 192	8, 192	14	14	1	0				
GI OB	CTL	CSA	N	228	0	0	0	1	0	0				
OSWA	CTL	CSA	N	228	8, 192	8, 192	7	7	1	0				
GOWA	CTL	CSA	N	228	0	0	0	1	0	0				
PST	CTL	CSA	N	231	53, 248	53, 248	13	1	13	0	GBL	INT	PST	
DPST	CTL	CSA	N	231	8, 192	8, 192	2	2	2	0	DEP	PST		
SAP	CTL	CSA	N	231	12, 288	12, 288	84	86	2	0				
GQMW	CTL	CSA	N	231	8, 192	8, 192	24	2	2	0				
LQMW	CTL	PRI	N	0	16, 384	16, 384	48	1	4	0				
D1WA	CTL	CSA	N	228	4, 096	4, 096	15	17	1	0				
DL2W	CTL	PRI	N	0	8, 192	8, 192	26	26	2	0				
DG2W	CTL	CSA	N	231	4, 096	4, 096	13	2	1	0				
QSAV	CTL	CSA	N	231	8, 192	8, 192	7	1	1	0				
VRPL	CTL	CSA	N	231	8, 192	8, 192	16	18	2	0				
LSAV	CTL	PRI	N	0	28, 672	28, 672	392	1	6	0				
					CSA	PRI VATE	EXT- CSA	EXT- PRI V						
CONTROL	330, 184				57, 344	0	0							
DLI SAS	0				0	0	0							

**Description:** This display shows summary information for all CBT pools and detail information for up to 15 pools per screen.

**Select Code:** POOLC

**Parameter:** Pool selection:

blanks      Enter four blanks to display all pools, beginning with the first 15 pools.

poolid      Enter the valid four-character pool ID of the first pool to be displayed. Pool IDs are defined in the IBM IMS Diagnosis Guide and Reference manual. When POOLC is requested, it displays a scrollable list of pools. The pool ID in the PARM field is the identifier of the next list of pools to be displayed. Pressing ENTER pages through the pool IDs and puts the pool ID shown in the PARM field at the top of the list.

**Field Descriptions:** Each of the fields is shown and described below by display area.

**Area 1**

ID	RGN	LOC	FX	SP	SIZE	HWM	BLK	BLKTH	GETM	FREEM	COMMENTS
I OSB	CTL	CSA	N	228	8, 192	8, 192	14	14	1	0	
GI OB	CTL	CSA	N	228	0	0	0	1	0	0	
OSWA	CTL	CSA	N	228	8, 192	8, 192	7	7	1	0	
GOWA	CTL	CSA	N	228	0	0	0	1	0	0	
PST	CTL	CSA	N	231	53, 248	53, 248	13	1	13	0	GBL INT PST
DPST	CTL	CSA	N	231	8, 192	8, 192	2	2	2	0	DEP PST
SAP	CTL	CSA	N	231	12, 288	12, 288	84	86	2	0	
GQMW	CTL	CSA	N	231	8, 192	8, 192	24	2	2	0	
LQMW	CTL	PRI	N	0	16, 384	16, 384	48	1	4	0	
D1WA	CTL	CSA	N	228	4, 096	4, 096	15	17	1	0	
DL2W	CTL	PRI	N	0	8, 192	8, 192	26	26	2	0	
DG2W	CTL	CSA	N	231	4, 096	4, 096	13	2	1	0	
QSAV	CTL	CSA	N	231	8, 192	8, 192	7	1	1	0	
VRPL	CTL	CSA	N	231	8, 192	8, 192	16	18	2	0	
LSAV	CTL	PRI	N	0	28, 672	28, 672	392	1	6	0	

This area shows the pool identification and size information. The descriptions are arranged in alphabetical order.

**BLK** Total number of control blocks in the pool.

**BLKTH** The threshold's target block number for STM (storage manager) task compression.

**COMMENTS** Additional significant information about the pool, usually a pool description.

**FREEM** Number of FREEMAINS issued for pool compression.

**FX** Indication whether pool is page-fixed (Y), not page-fixed (N), or conditionally page-fixed (C).

**GETM** Number of GETMAINs issued for pool expansion.

**HWM** High-water mark of the pool size in bytes.

**ID** The name of the CBT pool as defined in the IBM IMS *System Definition Reference* manual.

**LOC** The location of pool storage (CSA, ECSA, EPRIV, or PRIV).

**RGN** The region that owns the pool, such as CTL or DLS.

**SIZE** The size of the pool in bytes.

**SP** The number of the OS/390 subpool where the IMS pool is located.

<b><u>Area 2</u></b>				
	CSA	PRI VATE	EXT- CSA	EXT- PRI V
CONTROL	330, 184	57, 344	0	0
DLI SAS	0	0	0	0

This area shows storage size by region. The descriptions are arranged in alphabetical order.

CONTROL      The total CBT pool storage size held by the control region, shown by pool storage location.

DLISAS        The total CBT pool storage size held by the DL/I SAS region, shown by pool storage location.

## POOLS - Pool Summary (Non-CBT Variable and Fixed Pools)

BMC SOFTWARE ----- POOL SUMMARY (NON-CBT) ----- PERFORMANCE MGMT												
SERV ==> POOLS INPUT 14:50:18 INTVL=> 3 LOG=> N TGT=> IMSA												
PARM ==> , SORT=%C ROW 1 OF 16 SCROLL=> CSR												
EXPAND: MON(INTNL), LINESEL(DPOOL/QUEST/MFSST)												
CSA EXT-CSA PRIVATE EXT-PRIV												
CONTROL REG 17,976 304,720 0 283,952												
DLISAS REG 0 1,112 0 2,320												
NAME	TYP	RGN	LOC	SP	CURR	%CU	HI-MRK	%HI	MAX-SZ	WASTED	EXP+CMP	EXP-CMP
-----												
DPSB/PSBD	VAR	DLS	EPRV	0	2320	5	2320	5	40960	N/A	N/A	N/A
DLMP/PSBC	VAR	DLS	ECSA	231	424	3	424	3	12288	N/A	N/A	N/A
LUMP	FIX	CTL	ECSA	0	234624	2	234624	2	10000K	5691	4	4
DLDP/DMBP	VAR	DLS	ECSA	231	688	1	688	1	49152	N/A	N/A	N/A
AOIP	FIX	CTL	EPRV	0	22912	0	40120	0	NOLIM	708	5	1
CESS	FIX	CTL	EPRV	231	0	0	8760	0	NOLIM	87	4	0
CIOP	FIX	CTL	CSA	0	17976	0	17976	0	NOLIM	22162	0	0
DBWP/DMBW	VAR	DLS	ECSA	231	0	0	672	2	24576	N/A	N/A	N/A
EMHB	FIX	CTL	EPRV	231	0	0	0	0	NOLIM	0	0	0
EPCB	VAR	CTL	ECSA	231	0	0	0	0	12288	N/A	N/A	N/A
FPWP	FIX	CTL	EPRV	0	0	0	0	0	NOLIM	0	0	0
HIOP	FIX	CTL	EPRV	0	283952	0	283952	0	NOLIM	9613	8	6
LUMC	FIX	CTL	ECSA	231	70096	0	70096	0	10000K	42072	6	2
MFBP	VAR	CTL	EPRV	0	0	0	0	0	49152	N/A	N/A	N/A

### Description:

This display shows summary information for all non-CBT variable and fixed pools.

Up to 32 different buffer sizes can be used for a fixed pool. For each buffer size, you can specify the buffer count for the primary block and the buffer count for the secondary blocks. There is only one primary block per buffer size for a given fixed pool. When the space in the primary block is exhausted, a secondary block is allocated (expansion). Subsequent secondary blocks are allocated (expansion) as needed. You can specify an upper-limit size for a fixed pool.

#### Tuning Tips

- Secondary blocks (and potentially the primary block depending on your specification) can be deleted (compressed) when they are no longer used. In theory, expansions (allocation of secondary blocks) should occur only for a heavy workload. Performance would be degraded if new blocks are frequently allocated (expanded) and deleted (compressed). The primary blocks should handle most of the normal workload, while the secondary blocks handle a heavy workload.

If certain buffer sizes are rarely used (for example, once a week for some special processing), you can specify the primary blocks for those buffer sizes as compressible. This way, the primary blocks are allocated only when they are needed, as opposed to being allocated when the pool is initialized. Moreover, these blocks are deleted (compressed) when they are not needed. The DPOOL service can be used to determine the usage of all buffer sizes for a given fixed pool.

- Since the buffers in a fixed pool are of fixed sizes (there are 32 maximum buffer sizes allowed), some of the space in the buffers would be wasted. A small amount of wasted space is unavoidable. However, a large amount of wasted storage should be avoided because it can impact performance. The POOLS/ DPOOL services can be used to identify which pool and which buffer sizes have the problem.

#### Select Code:

POOLS

#### Parameter:

Multiple parameters must be separated by a comma (,).

Display sort:

**SORT**

You can enter:

, SORT|S0=cc

where cc can be any of the following two characters. The display list is sorted by %CU field (current usage percentage) by default. The sort sequence is ascending for alphanumeric characters and descending for numeric characters.

#### Tip

An easy way to sort is to move the cursor to the column heading and press ENTER.

The following SORT parameter descriptions are arranged alphabetically. SORT keywords that start with a special character are described first.

%C	Sorts the list by %CU (current usage percentage).
%H	Sorts the list by %HI (high-water mark percentage).
CU	Sorts the list by CURR (current size).
HI	Sorts the list by HI-MRK (high-water mark).
LO	Sorts the list by LOC (location).
MA	Sorts the list by MAX-SZ (maximum size).
NA	Sorts the list by NAME.
NE	Sorts the list by EXP-CMP (net expansions).
RG	Sorts the list by RGN (region).
SP	Sorts the list by SP (subpool).
TO	Sorts the list by EXP+CMP (total number of expansions and compressions).
TY	Sorts the list by TYP (type).
WA	Sorts the list by WASTED (accumulated wasted space for all buffer sizes for the pool since IMS started).

Pool selection:

TYPE	You can enter:
	TYPE=ALL   FIX   VAR
ALL	Displays all pools (default).
FIX	Displays only fixed pools.
VAR	Displays only variable pools.

**Expand:**

This POOLS display can be EXPANDED by moving the cursor to the following fields and pressing ENTER:

- MON(INTNL)  
Active Timer Requests display of all active IMS internals monitors (INTNL area).

- LINESEL(DPOOL/QUEST/MFSST)

The display is expanded to the appropriate detail for the first pool in the list:

- For an MFBP pool, the EXPAND is to the MFFST display service. This does not apply to an IMS DBCTL system.
- For a QBUF pool, the EXPAND is to the QUEST display service. This does not apply to an IMS DBCTL system.
- For all other pools, the EXPAND is to the DPOOL display service.

You can also move the cursor to any line in the display list and press ENTER to expand to the detail display for that pool.

**Sorting:**

The display list can be sorted by:

- Using the SORT parameter.
- Positioning the cursor with the TAB key to the column heading to be sorted and pressing ENTER.

This overrides any SORT parameter entered in the PARM field and primes the field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by %CU field (current usage percentage) by default. Alphanumeric fields are sorted in ascending order and numeric fields are sorted in descending order.

**Scrolling:**

The display is scrollable.

**Field Descriptions:**

Each of the fields is shown and described below by display area.

<b><u>Area 1</u></b>					
		CSA	EXT- CSA	PRIVATE	EXT- PRIV
CONTROL	REG	17, 976	304, 720	0	283, 952
DLISAS	REG	0	1, 112	0	2, 320

This area shows all storage usage by pools and by location.

**CONTROL REG and DLISAS REG**

The total allocated pool space in CSA, extended CSA, private, and extended private under each region.

**Area 2**

NAME	TYP	RGN	LOC	SP	CURR	%CU	HI - MRK	%HI	MAX-SZ	WASTED	EXP+CMP	EXP- CMP
DPSB/PSBD	VAR	DLS	EPRV	0	2320	5	2320	5	40960	N/A	N/A	N/A
DLMP/PSBC	VAR	DLS	ECSA	231	424	3	424	3	12288	N/A	N/A	N/A
LUMP	FIX	CTL	ECSA	0	234624	2	234624	2	10000K	5691	4	4
DLDP/DMBP	VAR	DLS	ECSA	231	688	1	688	1	49152	N/A	N/A	N/A
AOIP	FIX	CTL	EPRV	0	22912	0	40120	0	NOLIM	708	5	1
CESS	FIX	CTL	EPRV	231	0	0	8760	0	NOLIM	87	4	0
CIOP	FIX	CTL	CSA	0	17976	0	17976	0	NOLIM	22162	0	0
DBWP/DMBW	VAR	DLS	ECSA	231	0	0	672	2	24576	N/A	N/A	N/A
EMHB	FIX	CTL	EPRV	231	0	0	0	0	NOLIM	0	0	0
EPCB	VAR	CTL	ECSA	231	0	0	0	0	12288	N/A	N/A	N/A
FPWP	FIX	CTL	EPRV	0	0	0	0	0	NOLIM	0	0	0
HIOP	FIX	CTL	EPRV	0	283952	0	283952	0	NOLIM	9613	8	6
LUMC	FIX	CTL	ECSA	231	70096	0	70096	0	10000K	42072	6	2
MFBP	VAR	CTL	EPRV	0	0	0	0	0	49152	N/A	N/A	N/A

This area summarizes all pool allocation. The descriptions are arranged in alphabetical order.

**%CU**

The current allocation size percentage:

$$\text{CURR} / \text{MAX-SZ} * 100$$

This value is 0 when the displayed pool has no limit.

**%HI**

The high-water mark percentage:

$$\text{HI - MRK} / \text{MAX-SZ} * 100$$

This value is 0 when the displayed pool has no limit.

**CURR**

The current total pool size allocation.

**EXP+CMP**

The total number of block expansions and compressions for the pool. It applies only to fixed pools.

**Tuning Tip**

A high number could indicate some of the primary/secondary buffer allocations are too small. Use the DPOOL service to determine which buffer sizes have the problem.

**EXP-CMP**

The difference of expansions and compressions (net expansions) of blocks for the pool. It applies only to fixed pools.

**Tuning Tip**

A high number could indicate some of the primary/secondary buffer allocations are too small. Primary blocks should be able to handle most workloads. Use the DPOOL service to determine which buffer sizes have the problem.

**HI-MRK**

The high-water mark allocation size for both variable and fixed pools since IMS startup.

**LOC**

The storage location of the pool can be:

**CSA**

Common storage area

**ECSA**

Extended CSA

**EPRV**

Extended private

**PRV**

Private

**MAX-SZ**

The total pool size in bytes. The fixed pool can have unlimited size, which is displayed as NOLIM.

**NAME**

The pool identification can be:

pppp/dddd

or

pppp

where:

dddd      Is a pool descriptor.

pppp      Is the valid IMS pool ID.

Pool IDs are defined in the IBM, *IMS System Definition Reference* manual.

The descriptor added to the pool ID in the POOLS display is a more common term than the pool ID.

**AOIP**

A fixed automated operator interface buffer pool.

**CESS**

A fixed external subsystem work pool.

**CIOP**

A fixed pool for terminal buffers.

#### DBWP/DMBW

The DBWP (DMBW) pool is the pool for the DMB work areas.

#### DLDP/DMBP

The DLDP (DMBP) pool holds DMBs.

#### DLMP/PSBC

The DLMP (PSBC) pool is the total PSB pool in the:

- Control region (CTL) when the IMS option LSO is not S.
- CSA partition of the PSB pool when LSO=S.

#### DPSB/PSBD

The DPSB pool is the DL/I SAS private partition of the PSB pool.  
This appears only when LSO=S.

#### EMHB

A fixed pool for IFP message-driven regions.

#### EPCB

This pool is required for Fast Path database access.

#### FPWP

Fast Path fixed work pool.

#### HIOP

A fixed communication I/O buffer pool.

#### LUMC

A fixed pool for the APPC/IMS LU6.2 function.

#### LUMP

A fixed pool for the APPC/IMS LU6.2 function.

#### MAIN/WKAP

The MAIN (WKAP) pool is the general work pool.

#### MFBP

MFS pool.

#### PSBW

PSB work pool.

#### QBUF

QMGR buffer pool.

#### RGN

The ownership of the pool by region:

##### CTL

Control region.

##### DLS

DLISAS address space.

SP

The OS/390 subpool number where this pool is allocated.

TYP

The type of pool:

FIX

Fixed pools.

An asterisk indicates at least one overflow buffer is allocated.

VAR

Variable pools.

WASTED

The accumulated wasted space for all the buffer sizes in the pool since IMS started. It applies only to fixed pools.

**Tuning Tip**

Since the buffers in a fixed pool are of fixed sizes (there are 32 maximum buffer sizes allowed), some of the space in the buffers would be wasted. A small amount of wasted space is unavoidable. However, a large amount of wasted storage should be avoided because it can impact performance. If the value is large, use the DPOOL service to determine which buffer sizes can be adjusted (if any). If a buffer size uses only a few buffers (for example, 1), the primary buffer count per block can be reduced.

## Chapter 15. OS/390 System Displays

This chapter describes the displays that show the status, activity, and performance of IMS resources when IMS interacts with OS/390 and the resources under OS/390 control.

### DSPST - Dispatcher Statistics

BMC SOFTWARE ----- DISPATCHER STATISTICS -----				----- PERFORMANCE MGMT			
SERV ==>	DSPST	INPUT	13:42:27	INTVL=>	3	LOG=>	N
PARM ==>	RA0600I	- ENTER FIRST REGION NUMBER				TGT==>	I7A331CT
						SCROLL=>	N/A
*****OS DISPATCHER*****				*GOAL MODE ACTIVE*			
REGION	TYPE	ASID	DSP	SRV. UNITS	TCB	CPU	
	BBI - SS	15	195	12,413		13	
0	CONTROL	33	184	219,421		18	
0	DLI S	123	178	11,345		0	
1	DBT	62	180	293,161		20	
2	MPP	42	185	93,227		17	
3	BMP	101	190	81,522		12	
*****IMS DISPATCHER*****							
3 DYNAMIC SAPS							
3 DEPENDENT REGIONS							
1	ACTIVE BMP	0	ACTIVE MDP				
1	ACTIVE MPP	0	ACTIVE NDP				
1	ACTIVE DBT	0	ACTIVE FPU				
				317 TOTAL ITASKS DISPATCHED			

**Description:** This display summarizes OS/390 and IMS dispatching. It shows the status of the IMS regions as seen by OS/390 and provides statistics about internal IMS multitasking.

**Select Code:** DSPST

**Parameter:** No entry or enter 1 to display the control region and dependent regions 1 through 8.

Enter n to display the control region and dependent regions n through n+7, where n is a valid region identification number within the range of defined regions.

This service shows eight regions at a time. Changing the region ID number in the PARM field displays another range of eight regions.

**Field Descriptions:** Each of the fields is shown and described below by display area.

Area 1									
*****OS DISPATCHER*****									
REGION	TYPE	ASID	DSP	SRV. UNITS	TCB CPU	DOMAIN	PFGR	PFPD	
	BBI-SS	15	195	12, 413	13	72	108	1	
0	CONTROL	33	184	219, 421	18	72	108	1	
0	DLIS	123	178	11, 345	0	72	308	1	
1	DBT	62	180	293, 161	20	71	307	1	
2	MPP	42	185	93, 227	17	71	307	1	
3	BMP	101	190	81, 522	12	71	309	1	

This area displays OS/390 data about the IMS control and dependent regions. The descriptions are arranged in alphabetical order.

**Note:** If \*GOAL MODE ACTIVE\* is in the upper-right portion of this area of the display, SRM information does not apply and the DOMAIN, PFGR, and PFPD fields are not displayed.

ASID	OS/390 address space identification.
DOMAIN	SRM domain.
DSP	Current priority of the address space on the dispatch queue when this address space is swapped in.
PFGR	SRM performance group.
PFPD	SRM performance period.
REGION	Region ID, 0 for the control region and DL/1 SAS region, 1-31 for dependent regions.
SRV.UNITS	Total weighted service units for this address space.
TCB CPU	Total region TCB CPU time expressed in seconds.
TYPE	Region type:

BBI-SS	BBI-SS product address space
BMP	Batch message processing
BMP-WFI	BMP wait-for-input
CONTROL	Control region
DBT	DBCTL CICS thread
FPU	Fast Path utility
JBP	Java batch message processing
JMP	Java message processing
JMP-WFI	JMP wait-for-input
MDP	Message-driven Fast Path
MPP	Message processing
MPP-WFI	MPP wait-for-input
NDP	Non-message-driven Fast Path
ODB	DBCTL ODBA thread

#### Area 2

```
*****IMS DISPATCHER*****  
      3 DYNAMIC SAPS
```

This area shows the number of save area prefixes (SAPs). SAPs are used to control the internal multitasking of IMS ITASKS. A SAP display SAP is needed for each active event control block (ECB) or ITASK. IMS preassigns SAPs for logging functions or the dependent regions for example.

#### DYNAMIC SAPS

SAPs are set up to control communications I/O processing. The number depends on the MAXIO statement in the IMS generation or on the SAV parameter specification at execution time (this limits the amount of concurrent communication-related processing.) A differentiation is made between privileged and nonprivileged ITASKS. If selective dispatching becomes necessary (for example, if there is a CIOP pool space shortage), only privileged ITASKS are assigned a SAP and dispatched until the condition clears.

#### Area 3

```
      3 DEPENDENT REGIONS  
1 ACTIVE BMP      0 ACTIVE MDP  
1 ACTIVE MPP      0 ACTIVE NDP  
1 ACTIVE DBT      0 ACTIVE FPU
```

This area displays the current activity in the IMS system. The descriptions are arranged in alphabetical order.

#### ACTIVE BMP

The number of BMP and JMP regions scheduled currently.

#### ACTIVE DBT

The number of active DBCTL CICS and ODBA threads. ODBA threads are identified as ODB type regions in the OS DISPATCHER section of DSPST.

#### ACTIVE MPP

The number of MPP and JMP regions that are scheduled currently (idle message regions are not included).

#### ACTIVE FPU, ACTIVE MDP, ACTIVE NDP

If Fast Path is installed, these fields contain the number of FPU (Fast Path utility), MDP (message-driven program), and NDP (nonmessage-driven program) regions scheduled currently.

#### DEPENDENT REGIONS

The number of regions started.

#### Area 4

```
317 TOTAL ITASKS DISPATCHED
```

This area shows the work performed by the IMS dispatcher since restart:

#### TOTAL ITASKS DISPATCHED

The number of ITASKs dispatched.

**Note:** The difference between ITASKS created and ITASKs dispatched is approximately the number of IWAITs.

## RS - Real Storage

BMC SOFTWARE -----				REAL STORAGE			----- PERFORMANCE MGMT			
SERV ==> RS		INPUT		12: 30: 50	INTVL=> 3	LOG=> N	TGT==> IMSxxx			
PARM ==>		SCROLL=> N/A								
	TOTAL	AVAIL	LOCAL	LSQA	CSA	LPA	SQA			
TOTAL	258302	1086	225166	21654	15015	2466	14569			
PAGEABLE	194049		198763		14573	2367				
FIXED	63167		26403	21654	442	99	14569			
FIXEDB	943		924							
IMS ADDRESS SPACES				TOTAL	LSQA	NLSQA	PAGE	FIXED	FIXB	SLOTS
IMS CONTROL				1680	100	112	1468	212	99	
DBRC				365	35	2	328	37		
DL/I SAS				445	43	52	350	95	2	
REGION	TYPE	PROGRAM	TRANSACT	TOTAL	LSQA	NLSQA	PAGE	FIXED	FIXB	SLOTS
1	MDP	DFSIVP4		123	32	2	89	34		
2	MPP			120	33	2	85	35		

**Description:** This service summarizes the usage of page frames in the total system by the IMS control region and nine dependent regions.

**Select Code:** RS

**Parameter:** No entry or enter 1 to display dependent regions 1 through 9.

Enter n to display dependent regions n through n+8, where n is a valid region identification number within the range of defined regions.

This service shows nine regions at a time. Changing the region ID number in the PARM field displays another range of nine regions.

**Field Descriptions:** Each of the fields is shown and described below by display area.

Area 1							
	TOTAL	AVAIL	LOCAL	LSQA	CSA	LPA	SQA
TOTAL	258302	1086	225166	21654	15015	2466	14569
PAGEABLE	194049		198763		14573	2367	
FIXED	63167		26403	21654	442	99	14569
FIXEDB	943		924				

This area shows the usage of the real storage page frames in the system. It is a matrix where each line represents a state the frame is in: TOTAL, PAGEABLE, FIXED, and FIXEDB. Each column indicates page frame usage.

The descriptions are arranged in alphabetical order by page frame state (row) then by page frame usage (column).

### Page Frame State:

FIXED Number of fixed page frames in the system.

FIXEDB Fixed page frames below the 16MB line.

PAGEABLE	Number of pageable page frames in the system.
TOTAL	Total number of pageable, fixed, and long-term fixed page frames in the system, excluding the OS/390 nucleus.

**Page Frame Usage:**

AVAIL	Number of currently unassigned page frames that can be used.
CSA	Number of page frames used by the common service area (CSA).
LPA	Number of page frames used by the link pack area (LPA).
LOCAL	For regions running <code>VI RTUAL=VI RTUAL</code> , this column shows the number of page frames currently used by the working set.
LSQA	Number of page frames used by the local system queue area (LSQA).
SQA	Number of page frames used by the system queue area (SQA).
TOTAL	The total number of page frames that can be used.

<u>Area 2</u>								
IMS ADDRESS SPACES	TOTAL	LSQA	NLSQA	PAGE	FIXED	FIXB	SLOTS	
IMS CONTROL	1680	100	112	1468	212	99		
DBRC	365	35	2	328	37			
DL/I SAS	445	43	52	350	95	2		

This area shows page frame usage by the IMS and DB control and DL/I SAS regions. The descriptions are arranged in alphabetical order.

**FIXB**  
Number of fixed page frames below the 16MB line currently used by the working set for regions running `VI RTUAL=VI RTUAL`.

**FIXED**  
Number of short- and long-term fixed page frames currently used by the working set for regions running `VI RTUAL=VI RTUAL`.

**IMS ADDRESS SPACES**  
Indicates the page frames used by the:

DBRC	DB control region
DLI/SAS	DL/I SAS region
IMS control	IMS control region

**LSQA**  
Number of LSQA page frames required by the operating system for the dependent region.

**NLSQA**

Non-LSQA page frames required by the operating system for the dependent region.

**PAGE**

Number of pageable page frames currently used by the working set for regions running VI RTUAL=VI RTUAL.

**SLOTS**

Number of slots reserved on the page data sets for the dependent region.

**TOTAL**

Total number of page frames used by the dependent region. This includes both local and LSQA page frames.

Area 3

REGION	TYPE	PROGRAM	TRANSACT	TOTAL	LSQA	NLSQA	PAGE	FIXED	FIXB	SLOTS
1	MDP	DFSIVP4		123	32	2	89	34		
2	MPP			120	33	2	85	35		

This area summarizes the usage of real storage page frames for the first nine IMS dependent regions. The descriptions are arranged in alphabetical order.

**FIXB**

Number of fixed page frames below the 16MB line currently used by the working set for regions running VI RTUAL=VI RTUAL.

**FIXED**

Number of short- and long-term fixed page frames currently used by the working set for regions running VI RTUAL=VI RTUAL.

**LSQA**

Number of LSQA page frames required by the operating system for the dependent region.

**NLSQA**

Non-LSQA page frames required by the operating system for the dependent region.

**PAGE**

Number of pageable page frames currently used by the working set for regions running VI RTUAL=VI RTUAL.

**PROGRAM**

Contains the PSB name, if scheduled.

**REGION**

Region ID.

**SLOTS**

Number of slots reserved on the page data sets for the dependent region.

**TOTAL**

Total number of page frames used by the dependent region. This includes both local and LSQA page frames.

## TRANSACTION

Contains the transaction code if the transaction is scheduled.

## TYPE

Region type:

BMP	Batch message processing
DBT	DBCTL CICS thread
FPU	Fast Path utility
JBP	Java batch message processing
JMP	Java message processing
MDP	Message-driven Fast Path
MPP	Message processing
NDP	Non-message-driven Fast Path
ODB	DBCTL ODBA thread

# Chapter 16. IRLM Displays

This chapter describes services that analyze IMS database locking when IRLM is used.

## IRLM - IRLM IMS Status (IRLM 1.5)

BMC SOFTWARE		IRLM IMS STATUS		PERFORMANCE MGMT	
SERV ==> IRLM	INPUT	14:16:10	INTVL=> 3	LOG=> N	TGT==> IMSA
PARM ==>	LINE 0001 OF 0018 SCROLL=> CSR				
SSID: IR15(2)	REL: 150	SYSTEM ACTIVITY:		REQUEST COUNTERS:	
PC(XMEM).....YES		GLOBAL LOCK.....2		IDENTIFY.....1	
RULES.....COMPAT		PTB LOCK.....0		LOCK.....59	
SCOPE.....LOCAL		RH LOCK.....0		UNLOCK.....49	
		RH NOTIFY.....0		SNAP.....0	
				CHANGE.....0	
GLOBAL STATUS:		EXIT ROUTINES CALLED:		SYNC NOTIFY.....1	
INTERNAL TRACE ACTIVE		SUSPEND.....0		ASYNCTIFY.....0	
		RESUME.....0		VERIFY.....0	
		STATUS.....0		PURGE.....1	
		NOTIFY.....0		QUERY.....0	
IMS STATUS:		DEADLOCK.....0		QUIT.....0	
NORMAL		TIMEOUT.....0		TAKEOVER.....0	
		TYPE:		RATIOS:	
		UPDATE		SUSPEND/LOCK.....0.00%	
		SHARE		DEADLOCK/SUSPEND...0.00%	
		PRIMARY		PTB-LOCK/LOCK.....0.00%	
***** END OF DISPLAY *****					

- Description:** The IRLM service shows statistics and status information from the IRLM to which the target IMS is identified. Request counters, IMS wait counts, and status are for the target IMS only. IRLM contention and exit routine counts are for all IMSs using this IRLM.
- Select Code:** IRLM
- Parameter:** None
- Scrolling:** This display is scrollable.
- Field Descriptions:** Each of the fields is shown and described below by display area.

<b>Area 1</b>	
SSID: IRLM(1)	REL: 150
PC(XMEM).....YES	
RULES.....COMPAT	
SCOPE.....LOCAL	

This area shows IRLM statistics. The descriptions are arranged in alphabetical order.

PC(XMEM)

YES or NO (specified in DXRJPROC) defines whether IRLM uses the cross-memory program call (PC) service.

REL:

The IRLM release number (for example, 150).

RULES

AVAIL or COMPAT (specified in DXRJPROC) defines the recovery procedure for this IRLM when it is the surviving partner in an interprocessor data sharing complex.

SCOPE

Specified in DXRJPROC.

LOCAL

Indicates that IRLM can only be used as the local lock manager (all sharing subsystems must use the same IRLM on the same OS/390).

GLOBAL

Indicates that the IRLM can participate with other IRLMs in an intraprocessor or interprocessor environment.

SSID:

The IRLM subsystem name and ID (specified in the IRLM startup procedure DXRJPROC with the parameters IRLNM and IRLMID).

**Area 2**

```
SYSTEM ACTIVITY:
GLOBAL LOCK. . . . . 2
PTB LOCK. . . . . 0
RH LOCK. . . . . 0
RH NOTI FY. . . . . 0
```

The fields in this area display system activity. The descriptions are arranged in alphabetical order.

GLOBAL LOCK

Total global lock requests.

PTB LOCK

Total RH (Request Handler) to PTB (Pass The Buck) lock requests.

RH LOCK

RH (Request Handler) to RH lock requests.

RH NOTIFY

RH (Request Handler) to RH notify requests.

### Area 3

SYNC NOTIFY.....1

#### REQUEST COUNTERS:

IDENTIFY.....	1
LOCK.....	59
UNLOCK.....	49
SNAP.....	0
CHANGE.....	0
ASYNC NOTIFY.....	0
VERIFY.....	0
PURGE.....	1
QUERY.....	0
QUIT.....	0
TAKEOVER.....	0

This area shows the number of requests from the target IMS to this IRLM. The descriptions are in alphabetical order.

#### ASYNC NOTIFY

The number of asynchronous notifies from this subsystem to other subsystems (database buffer invalidate or extension and I/O error for example).

#### CHANGE

Number of lock state change requests by this subsystem.

#### IDENTIFY

Number of times this subsystem identified itself to IRLM.

#### LOCK

Number of resource lock requests by this subsystem.

#### PURGE

Number of times this subsystem requested IRLM to purge its locks or responded to an IRLM status exit.

#### QUERY

Number of query requests by this subsystem.

#### QUIT

Number of requests by this subsystem to quit using IRLM.

#### SNAP

Number of requests by this subsystem to SNAP IRLM storage.

#### SYNC NOTIFY

Number of synchronous notify requests by this subsystem.

#### TAKEOVER

Number of XRF backup requests to take over processing.

#### UNLOCK

Number of resource unlock requests by this subsystem.

#### VERIFY

Number of times this subsystem requested IRLM to verify the status of other subsystems with which it shares databases.

#### Area 4

GLOBAL STATUS:  
INTERNAL TRACE ACTIVE

This area shows the status of IRLM global mode. The descriptions are arranged in alphabetical order.

The following status messages can be displayed:

#### COMMUNICATION FAILURE

IRLM enters this state after an operator issues **SETSTATE, COMM**. The processing IRLM tells its IMS to quiesce all activity against all global block level shared databases to preserve data integrity since it is unsure of the status of the other IRLM. The processing IRLM prevents further transaction scheduling until an operator issues **SETSTATE, INIT** to set the INITIAL state.

#### INTERNAL TRACE (NOT) ACTIVE

Internal trace is (not) active. This status always appears.

#### SYSTEM FAILURE

IRLM enters this state after an operator issues **SETSTATE, SYSTEM**, which means its partner IRLM has failed. This IRLM keeps the locks of the failing IRLM on the potentially updated but uncommitted data. To ensure data integrity, this data cannot be accessed. Access to all other shared resources is still permitted. IRLM goes from this state to INITIAL state when the failed IRLM comes back to report it has completed its cleanup or when the retained locks are purged manually.

#### TAKEOVER IN PROGRESS

An XRF (Extended Recovery Facility) takeover is in progress.

#### Area 5

EXIT ROUTINES CALLED:  
SUSPEND. .... 0  
RESUME. .... 0  
STATUS. .... 0  
NOTIFY. .... 0  
DEADLOCK. .... 0  
TIMEOUT. .... 0

This area shows the number of times an exit routine is called by this IRLM for any IMS. The descriptions are arranged in alphabetical order.

#### DEADLOCK

This exit is entered by IRLM during a deadlock detection cycle if an IMS is involved in a deadlock. DFSDLKX0 calculates a worth value for each of its work units involved in the deadlock. IRLM then uses the value to select a candidate to break the deadlock.

#### NOTIFY

IRLM drives the notify exit to provide notification of certain events. For IMS, it calls DFSNOTX0 to handle these events. An example of an event requiring notification is data set extension by another IMS.

#### RESUME

IRLM calls the resume exit when a work unit can access the lock it is waiting for. For IMS, it calls DFSRESX0 to IPOST the work unit (ITASK) out of IWAIT.

#### STATUS

IRLM drives the status exit when it is told about a communication failure or a system failure of the partner IRLM.

#### SUSPEND

IRLM calls the suspend exit when a work unit must wait for a lock. For IMS, it calls DFSSUSX0 to IWAIT the work unit (ITASK).

#### **Area 6**

IMS STATUS:  
NORMAL

This area shows the status of the target IMS in response to the activity of its partner IRLM. The descriptions are arranged in alphabetical order.

#### NORMAL

This IMS is in a normal state which can be qualified if any of the other status messages appear.

#### ONLY ALLOW QUIT

No other request for this IMS is accepted except quit.

#### PURGE REQUIRED

QUIT HELD=RETAIN was issued to keep all the locks held by this IMS. A purge is required to release the locks.

#### QUIT ISSUED

A quit was issued for this IMS. Any work unit still in progress within this IMS is cancelled. All locks held by this IMS are released if HELD=RELEASE is issued or retained if HELD=RETAIN is issued. This information also is sent to the other IRLM in the data sharing group to update its ISL (Identified Subsystem List).

#### SUBSYS LOCKS RETAINED

A QUIT HELD=RETAIN was issued for this IMS to stop all of its work in progress but to keep all its locks. This is usually an abnormal state (something is wrong with this IMS or its related IRLM).

#### Area 7

TYPE:  
UPDATE  
SHARE  
PRIMARY

This area displays the type of data sharing that this IMS does. The descriptions are arranged in alphabetical order.

#### BACKUP

IMS is the backup system in the XRF complex.

#### NO SHARE

IMS does not share its databases with other IMS environments.

#### PRIMARY

IMS is the primary system in the XRF complex. A non-XRF IMS is also considered primary.

#### READ ONLY

IMS can only read its databases.

#### SHARE

IMS shares its databases with other IMS environments.

#### UPDATE

IMS can update its databases.

#### Area 8

RATIO S:  
SUSPEND/LOCK. . . . . 0. 00%  
DEADLOCK/SUSPEND. . . . . 0. 00%  
PTB- LOCK/LOCK. . . . . 0. 00%

This area shows percentages of lock requests that are in suspension or a potential deadlock. The descriptions are arranged in alphabetical order.

#### DEADLOCK/SUSPEND

The percentage of suspends detected in a potential deadlock condition during a deadlock cycle. This ratio is computed as follows:

100 x calls to deadlock exit routines / calls to suspend exit

#### PTB-LOCK/LOCK

The percentage of lock requests that require a PTB process, computed as follows:

100 x PTB lock / lock results

#### SUSPEND/LOCK

The percentage of lock requests that result in the suspend state, computed as follows:

100 x calls to suspend exit / lock requests

## IRLMG - IRLM GLOBAL STATUS (IRLM 1.5)

```
BMC SOFTWARE ----- IRLM GLOBAL STATUS ----- PERFORMANCE MGMT
SERV ==> IRLMG          INPUT   14:16:17  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==>                                LINE 0001 OF 0030  SCROLL=> CSR

SSID: IR15(2)    REL: 150  SYSTEM ACTIVITY:      REQUEST COUNTERS:
PC(XMEM).....YES      GLOBAL LOCK..... 4  IDENTIFY..... 7
RULES.....COMPAT      PTB LOCK..... 0  LOCK..... 1,187
SCOPE.....LOCAL      RH LOCK..... 0  UNLOCK..... 560
                                RH NOTIFY..... 0  SNAP..... 0
                                CHANGE..... 18
GLOBAL STATUS:      EXIT ROUTINES CALLED:
INTERNAL TRACE ACTIVE  SUSPEND..... 26  SYNC NOTIFY..... 1
                                RESUME..... 24  ASYNC NOTIFY..... 0
                                STATUS..... 0  VERIFY..... 2
                                NOTIFY..... 0  PURGE..... 1
                                DEADLOCK..... 0  QUERY..... 0
                                TIMEOUT..... 0  QUIT..... 1
                                TAKEOVER..... 0

VTAM INFORMATION:
COMCYCL..... 500MS  DEADLOCK..... (15S, 4)  RATIOS:
VTAM SENDS..... 0  LOCAL..... 0  SUSPEND/LOCK..... 2.2%
AVG BUCK SIZE..... 0  GLOBAL..... 0  DEADLOCK/SUSPEND... 0.0%
                                MAXCSA..... 4,096,000  PTB-LOCK/LOCK..... 0.0%
                                CURRENT..... 11,392
                                HIGHEST..... 11,392

===== IDENTIFIED SUBSYSTEMS =====
SUBSYSID    STATE    TYPE    INTENT    OTHER STATUS
-----
DB2D        ACTIVE    DB2     UPDATE    NORMAL
            PRIMARY   LOCAL   NO SHARE
X16H        ACTIVE    ONLINE  UPDATE    PURGE ISSUED
            PRIMARY   LOCAL   SHARE
CIR1IRLM    ACTIVE    BATCH   UPDATE    NORMAL
            PRIMARY   REMOTE  SHARE
***** END OF DISPLAY *****
```

**Description:** The IRLMG service shows statistics and status information from the IRLM to which the target subsystem is identified. It includes information related to all subsystems using this IRLM.

**Select Code:** IRLMG

**Parameter:** None

**Scrolling:** This display is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

**Area 1**

```
SSID: 1R15(2)      REL: 150
PC(XMEM).....YES
RULES.....COMPAT
SCOPE.....LOCAL
```

This area shows IRLM statistics. The descriptions are arranged in alphabetical order.

**PC(XMEM)**

YES or NO (specified in DXRJPROC) defines whether IRLM uses the cross-memory program call (PC) service.

**REL:**

The IRLM release number.

**RULES**

AVAIL or COMPAT (specified in DXRJPROC) defines the recovery procedure for this IRLM when it is the surviving partner in an interprocessor data sharing complex.

**SCOPE**

Specified in DXRJPROC.

**LOCAL**

Indicates that IRLM can only be used as the local lock manager (all sharing subsystems must use the same IRLM on the same OS/390).

**GLOBAL**

Indicates that the IRLM can participate with other IRLMs in an intraprocessor or interprocessor environment.

**SSID:**

The IRLM subsystem name and ID (specified in the IRLM startup procedure DXRJPROC with the parameters IRLNM and IRLMID).

**Area 2**

```
SYSTEM ACTIVITY:
GLOBAL LOCK.....4
PTB LOCK.....0
RH LOCK.....0
RH NOTIFY.....0
```

The fields in this area display system activity. The descriptions are arranged in alphabetical order.

**GLOBAL LOCK**

Total global lock requests.

**PTB LOCK**

Total RH (Request Handler) to PTB (Pass The Buck) lock requests.

## RH LOCK

RH (Request Handler) to RH lock requests.

## RH NOTIFY

RH (Request Handler) to RH notify requests.

### Area 3

#### REQUEST COUNTERS:

IDENTIFY. ....	7
LOCK. ....	1, 187
UNLOCK. ....	560
SNAP. ....	0
CHANGE. ....	18
SYNC NOTIFY. ....	1
ASYNC NOTIFY. ....	0
VERIFY. ....	2
PURGE. ....	1
QUERY. ....	0
QUIT. ....	1
TAKEOVER. ....	0

This area shows the number of requests from any subsystem (IMS or CICS for example) to this IRLM. The descriptions are in alphabetical order.

## ASYNC NOTIFY

Number of asynchronous notifies from IRLM or the subsystems shown by this IRLMG service to other subsystems (database buffer invalidate, extension and I/O error, or IRLM failure for example).

## CHANGE

Number of lock state change requests by subsystems.

## IDENTIFY

Number of subsystem identifies to IRLM.

## LOCK

Number of resource lock requests by subsystems.

## PURGE

Number of times IRLM was requested to purge and the number of responses to IRLM status exits.

## QUERY

Number of query requests by subsystems.

## QUIT

Number of requests by subsystems to quit using IRLM.

## SNAP

Number of requests by subsystems to SNAP IRLM storage.

## SYNC NOTIFY

Number of synchronous notify requests by subsystems.

#### TAKEOVER

Number of XRF backup requests to take over processing.

#### UNLOCK

Number of resource unlock requests by subsystems.

#### VERIFY

Number of times IRLM was requested by a subsystem to verify the status of other subsystems sharing databases with the requesting subsystem.

#### **Area 4**

GLOBAL STATUS:  
INTERNAL TRACE ACTIVE

This area shows the status of IRLM global mode. The descriptions are arranged in alphabetical order.

The following status messages can be displayed:

#### COMMUNICATION FAILURE

IRLM enters this state after an operator issues SETSTATE, COMM. The processing IRLM tells its IMS to quiesce all activity against all global block level shared databases to preserve data integrity since it is unsure of the status of the other IRLM. The processing IRLM prevents further transaction scheduling until an operator issues SETSTATE, INIT to set the INITIAL state.

#### INTERNAL TRACE (NOT) ACTIVE

Internal trace is (not) active. This status always appears.

#### SYSTEM FAILURE

IRLM enters this state after an operator issues SETSTATE, SYSTEM, which means its partner IRLM has failed. This IRLM keeps the locks of the failing IRLM on the potentially updated but uncommitted data. To ensure data integrity, this data cannot be accessed. Access to all other shared resources is still permitted. IRLM goes from this state to INITIAL state when the failed IRLM comes back to report it has completed its cleanup or when the retained locks are purged manually.

#### TAKEOVER IN PROGRESS

An XRF (Extended Recovery Facility) takeover is in progress.

#### **Area 5**

```
EXIT ROUTINES CALLED:
SUSPEND. .... 26
RESUME. .... 24
STATUS. .... 0
NOTIFY. .... 0
DEADLOCK. .... 0
TIMEOUT. .... 0
```

This area shows the number of times an exit routine is called by this IRLM for any IMS. The descriptions are arranged in alphabetical order.

#### **DEADLOCK**

This exit is entered by IRLM during a deadlock detection cycle if an IMS is involved in a deadlock. DFSDLKX0 calculates a worth value for each of its work units involved in the deadlock. IRLM then uses the value to select a candidate to break the deadlock.

#### **NOTIFY**

IRLM drives the notify exit to provide notification of certain events. For IMS, it calls DFSNOTX0 to handle these events. An example of an event requiring notification is data set extension by another IMS.

#### **RESUME**

IRLM calls the resume exit when a work unit can access the lock it is waiting for. For IMS, it calls DFSRESX0 to IPOST the work unit (ITASK) out of IWAIT.

#### **STATUS**

IRLM drives the status exit when it is told about a communication failure or a system failure of the partner IRLM.

#### **SUSPEND**

IRLM calls the suspend exit when a work unit must wait for a lock. For IMS, it calls DFSSUSX0 to IWAIT the work unit (ITASK).

#### **Area 6**

```
VTAM INFORMATION:
COMCYCL. .... 500MS
VTAM SENDS. .... 0
AVG BUCK SIZE. .... 0
```

This area shows VTAM statistics for this partner IRLM of the target IMS. The descriptions are arranged in alphabetical order.

#### **APPLS**

The VTAM APPL names of this IRLM and its partner. It defines the primary session as specified in DXRJPROC APPLS. This field is displayed only for SCOPE=GLOBAL.

#### **APPL2**

The VTAM APPL names of this IRLM and its partner. It defines the alternative session between two IRLMs as specified in DXRJPROC APPL2. This field is displayed only for an XRF complex with SCOPE=GLOBAL. If APPL2 is specified, APPLS is not applicable.

### APPL3

The VTAM APPL names of this IRLM and its partner. It defines another backup session between two IRLMs as specified in DXRJPROC APPL3. This field is displayed only for an XRF complex with SCOPE=GLOBAL.

### AVG BUCK SIZE

The average size of the buck being passed between the IRLMs.

### COMCYCL

The time the IRLM delays before processing its inter-IRLM requests (specified in COMCYCL of the IRLM startup procedure DXRJPROC).

### VTAM SENDS

The total VTAM SENDs by this IRLM.

#### Area 7

```
DEADLOCK. . . . . (15S, 4)
LOCAL. . . . . 0
GLOBAL. . . . . 0
```

This area provides deadlock mode statistics. The descriptions are arranged in alphabetical order.

### DEADLOCK

The first number is the time in seconds between IRLM local deadlock cycles. The second number represents the number of local deadlock cycles before global deadlock detection is performed (defined in DEADLOCK of DXRJPROC).

### GLOBAL

Total number of global deadlocks.

### LOCAL

Total number of local deadlocks.

#### Area 8

```
MAXCSA. . . . . 4, 096, 000
CURRENT. . . . . 11, 392
HIGHEST. . . . . 11, 392
```

This area shows CSA/ECSA usage by IRLM. The descriptions are arranged in alphabetical order.

### CURRENT

Current CSA/ECSA usage.

### HIGHEST

CSA/ECSA usage high-water mark.

### MAXCSA

The maximum amount of CSA/ECSA that the IRLM can use for its dynamic control blocks (specified in MAXCSA of DXRJPROC).

### Area 9

#### RATIOS:

SUSPEND/LOCK. . . . . 2. 2%

DEADLOCK/SUSPEND. . . . . 0%

PTB- LOCK/LOCK. . . . . 0%

This area shows percentages of lock requests that are in suspension or a potential deadlock. The descriptions are arranged in alphabetical order.

#### DEADLOCK/SUSPEND

Percentage of suspends detected in a potential deadlock condition during a deadlock cycle. This ratio is computed as follows:

100 x calls to deadlock exit routines / calls to suspend exit

#### PTB-LOCK/LOCK

Percentage of lock requests that require a PTB process. This ratio is computed as follows:

100 x PTB lock / lock requests

#### SUSPEND/LOCK

Percentage of lock requests that result in suspend state. This ratio is computed as follows:

100 x calls to suspend exit / lock requests

### Area 10

```
===== IDENTIFIED SUBSYSTEMS =====
SUBSYSID   STATE   TYPE   INTENT   OTHER STATUS
-----
DB2D       ACTIVE  DB2    UPDATE   NORMAL
           PRIMARY LOCAL   NO SHARE
X16H       ACTIVE  ONLINE UPDATE   PURGE ISSUED
           PRIMARY LOCAL   SHARE
CIR11RLM   ACTIVE  BATCH  UPDATE   NORMAL
           PRIMARY REMOTE  SHARE
***** END OF DISPLAY *****
```

This area shows the subsystems that have identified themselves to IRLM. The descriptions are arranged in alphabetical order.

#### INTENT

The subsystem data sharing intent:

NO SHARE|SHARE

The subsystem either participates in data sharing or does not participate.

READ ONLY|UPDATE

The subsystem intent is update or read only.

## OTHER STATUS

Other subsystem status indicators:

### NORMAL

The subsystem state is normal and none of the following occur.

### PURGE ISSUED

The subsystem issued a PURGE to release all locks.

### QUIT ISSUED

The subsystem issued a quit to stop all of its work in progress.

### SUBSYS LOCKS RETAINED

A quit was issued with HELD=RETAIN to keep all of its locks on updated but uncommitted data.

## STATE

State of the subsystem:

### ACTIVE|INACTIVE

The subsystem is active or inactive.

### BACKUP|PRIMARY

The subsystem is the primary or the backup in the XRF complex. A non-XRF subsystem is also considered primary.

## SUBSYSID

Subsystem name. For online IMS or DB2, this is the 4-byte subsystem ID. For IMS batch and CICS local DL/I, this is the jobname or started task name.

## TYPE

Subsystem type:

### BATCH

The subsystem is IMS batch (including CICS local DL/I).

### DB2

The subsystem is DB2.

### LOCAL

The subsystem is connected to this IRLM (local).

### ONLINE

The subsystem is an IMS online system.

### REMOTE

The subsystem is connected to its partner IRLM (IMS block level sharing only).

# LCRES - IRLM Lock Contention by Resource (IRLM 1.5)

```
BMC SOFTWARE ----- IRLM LOCK CONT BY RESRC ----- PERFORMANCE MGMT
SERV ==> LCRES          INPUT    16: 56: 54  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==>                                LINE    1 OF    1  SCROLL=> CSR
                                           << EXPAND >>

----- LOCK----- CUR          WAIT
DATABASE DCB/AREA RBA/TYPE F G STA    HOLDERS    CNT    WAITERS
-----
CUSTHDAM          1 0000615C P G SHR  003-I16XBMPW  1  002-I16XBMPW
***** END OF DATA *****
```

- Description:** The LCRES service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of the locks related to the target IMS that are in lock contention (being waited for).
- Select Code:** LCRES
- Parameter:** None
- Expand:** You can use the cursor to select a pop-up scrollable window, which lists the work units in contention for the selected lock.
- Scrolling:** The list of locks is scrollable.
- Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

CUR STA  
Current state of the lock. The state can be:

EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

HOLDERS  
Identifies the holder of the lock on a resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.

X - xxxxxxxx      Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.

If the lock is being held by more than one region, the count of the holders is shown.

## LOCK

Resource waited on or held.

### DATABASE

Database name.

### DCB/AREA

DCB number of a full function DL/I database data set or the area name for a Fast Path DEDB.

### F

Fast Path indicator:

blank	Non-Fast Path lock
B	B-lock for share level greater than 2, DS-BUSY, DATASET, and COMMAND
F	Fast Path lock
N	Non-updatable Fast Path area lock
P	P-lock for share level greater than 2, EXTEND, and BID

### G

Global indicator:

blank	Local lock
G	Global lock

## RBA/TYPE

In case of an RID (record lock) or BID (block lock), this field contains an identifier IMS uses to control access to the resource. The rules IMS uses to construct the ID are listed below to help in problem determination.

### FP DEDB

The ID is the high-order three bytes of the RBA of the VSAM control interval containing the segment.

### FP MSDB

The ID is the address of the segment prefix in the IMS control region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

### HDAM OSAM

The ID is the RBA of the block containing the segment.

### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (X' 40' ) of the high order byte is turned on.

#### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

#### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

There are many types of locks to serialize various resources. The most common ones are block lock (BID) and record lock (RID). Block lock is used to serialize the update to a block or CI. Record lock is used to deny access to uncommitted data. In some cases, this service uses resource ID values that do not represent a relative byte address. In these cases, the following text is displayed in place of the actual ID:

#### COMMAND

Used for global commands, IDENTIFY to IRLM, or IRLM NOTIFY.

#### DATASET

Used at data set open for buffer invalidation notifies.

#### DS-BUSY

Data set busy. Used to ensure no CI or CA split during a read.

#### EXTEND

Used to serialize a data set extension.

The following locks are Fast Path only:

#### AREA

Used by Fast Path to lock an entire area. This is used mostly during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

#### DUMMY

Force release of all global locks.

#### NONDBRC

Database is not registered with DBRC.

#### WAIT CNT

Total number of waiters on the lock.

#### WAITERS

Identifies the first two waiters for that locked resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.

L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.

### Selected Lock Pop-Up

BMC SOFTWARE				IRLM LOCK CONT BY RESRC				PERFORMANCE MGMT			
SERV ==>		LCRES		INPUT		16: 57: 02		INTVL=>		3 LOG=> N TGT==> IMSA	
PARM ==>								LINE		1 OF 1 SCROLL=> CSR	
-----LOCK-----				CUR		WAIT					
DATABASE		DCB/AREA		RBA/TYPE		F G STA		HOLDERS		CNT WAITERS	
-----											
CUSTHDAM		1		0000615C		P G SHR		003-I16XBMPW		1 002-I16XBMPW	
=====											
-----HOLDERS-----						-----WAITERS-----				REQ	
RGN T		STC/JOB		PSB		STATUS		RGN T		STC/JOB PSB STAT	
-----											
003 B		I16XBMPW		PTEST01		ACTV-USR		002 B		I16XBMPW PTEST02 UPD	
***** END OF DATA *****											

This pop-up shows the work units in contention for the selected lock. It is an EXPAND of a lock by cursor selection from the LCRES service.

### HOLDERS

Lists all the holders of the lock.

### PSB

The program specification block that is being processed in the online IMS region.

### RGN

Region (PST) number of the online IMS region; otherwise, the field is left blank. The region number is prefixed by an \* character if the region belongs to another IMS.

### STATUS

Region status:

ACTIVE            Active in a nonspecific process.

ACTV-BKO        Region in dynamic backout.

ACTV-DB2        Active in DB2 (Event Collector must be active).

ACTV-DBR        Region active in DBRC.

ACTV-DLI        Region active in DL/I.

ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	<p>Region defined but not started (not yet signed on).</p> <p>The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.</p>
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	<p>Region waiting for an AO message.</p> <p>Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.</p>
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	<p>Region in scheduling (waiting for block mover latch).</p> <p><b>Note:</b> The block mover latch comprises several smaller latches. To find which BML latch a region is waiting for, use the LATCH service.</p>
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).

WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.

WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### STC/JOB

Name of the job or started task. For an online IMS on a partner IRLM, this is the subsystem ID of that IMS. For an IMS internal task, this is the 3-byte field task ID (for example, XFP for Fast Path internal task).

#### T

Region type. Possible types:

B	Batch message processing (BMP) or Java batch message processing (JBP) region	
D	DBCTL thread (CICS or ODBA)	
F	Fast Path region (IFP)	
L	Local batch IMS region	
M	Message processing (MPP) or Java message processing (JMP) region	
S	IMS internal task region	
T	TPI (explicit APPC program)	
X	External source (from a partner IRLM)	

#### WAITERS

Lists all the waiters for the lock.

REQ STAT	Requested lock state:	
	EXC	Exclusive
	RO	Read-only
	SHR	Share
	UNK	Unknown (lock state in transition)
	UPD	Update

# LCUSR - IRLM Lock Contention by User (IRLM 1.5)

BMC SOFTWARE ----- IRLM LOCK CONT BY USER ----- PERFORMANCE MGMT									
SERV ==> LCUSR		INPUT 14: 14: 54		INTVL=> 3		LOG=> N		TGT==> IMSA	
PARM ==>				LINE 1 OF		2		SCROLL=> CSR	
<< EXPAND >>									
-----USER-----				-----HOLDING-----				-- WAITING FOR--	
RGN	T	STC/JOB	PSB	TRANCODE	RO	SHR	UPD	EXC	WAITERS
-----									
001	B	I 16XBMPW	PTEST01		0	1	0	0	1
002	B	I 16XBMPW	PTEST02		0	0	2	0	0
001 L CIR11RLM									
***** END OF DATA *****									

**Description:** The LCUSR service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of the target IMS work units (dependent regions) that are involved in lock contentions (waiting or holding resources that someone is waiting for). The regions are displayed in sequence by region number.

The user causing a lock contention is highlighted.

**Select Code:** LCUSR

**Parameter:** None

**Expand:** You can select a detailed display (LUSRD) of any listed region, which identifies the resource(s) being waited for or held by that region. Position the cursor on the line for that region and press ENTER as indicated on the display by <<EXPAND>>. Pressing the PF3 (END) key from that display returns to this list.

**Scrolling:** The list of regions is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

## HOLDING

The locks being held by this region:

EXC	Number of EXCLUSIVE level locks being held
RO	Number of READ-ONLY level locks being held
SHR	Number of SHARE level locks being held
UPD	Number of UPDATE level locks being held

## USER

Identifies the target online IMS work units.

PSB                    The name of the PSB (program specification block) being processed in this online region.

RGN                    Region (PST) number.

STC/JOB                Name of the job or started task.

T	Region type. Possible types:
B	Batch message processing (BMP) or Java batch message processing (JBP) region
D	DBCTL thread (CICS or ODBA)
F	Fast Path region (IFP)
M	Message processing (MPP) or Java message processing (JMP) region
T	TPI (explicit APPC program)
TRANCODE	The transaction code being processed by the online region, if the application is message driven.

#### WAITERS

Number of other regions waiting for one or more locks held by this region.

#### WAITING FOR

The region that this region is waiting for. There can be more than one region waiting, but only one is listed here.

RGN	Region (PST) number. If the region is not an online IMS region, this field is blank. If the region belongs to another IMS (different subsystem ID), the region number is prefixed by an * character; for example, *001.
STC/JOB	The name of the job or started task. For an online IMS on a partner IRLM, this is the subsystem ID of that IMS. For an IMS internal process, this is the 3-byte IMS internal task field name; for example, XFP for Fast Path internal task.
T	Region type. Possible types:
B	Batch message processing (BMP) or Java batch message processing (JBP) region
D	DBCTL thread (CICS or ODBA)
F	Fast Path region (IFP)
L	Local batch IMS region
M	Message processing (MPP) or Java message processing (JMP) region
S	IMS internal task
T	TPI (explicit APPC program)
X	External source (from a partner IRLM)

# LHRES - IRLM Locks Held by Resources (IRLM 1.5)

BMC SOFTWARE ----- IRLM LOCKS HELD BY RESRC ----- PERFORMANCE MGMT									
SERV ==> LHRES		INPUT		16: 57: 09		INTVL=> 3		LOG=> N	
PARM ==>						LINE		1 OF 6	
								SCROLL=> CSR	
								<< EXPAND >>	
----- LOCK----- CUR									
DATABASE		DCB/AREA		RBA/TYPE		F G STA		HOLDERS	
								WAIT	
								CNT	
								WAITERS	
-----									
		COMMAND		B G RO		S- DLI			
		COMMAND		F G RO		S- DLI			
CUSTHDAM		1		DATASET		B G RO		S- DLI	
CUSTHDAM		1		0000490E		P G UPD		002- I 16XBMPW	
CUSTHDAM		1		0000615C		P G SHR		003- I 16XBMPW 1 002- I 16XBMPW	
DBFSAMD3		CUSDB		AREA		N G RO		S- XFP	
***** END OF DATA *****									

**Description:** The LHRES service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of all the locks related to the target IMS that are being held (in contention or not).

**Select Code:** LHRES

**Parameter:** None

**Expand:** You can use the cursor to select a pop-up scrollable window, which lists all the work units in contention for the selected lock.

**Scrolling:** The list of locks is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

## CUR STA

Current state of the lock. The state can be:

EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

## HOLDERS

Identifies the holder of the lock on a resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.

S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.
SHR CNT	If the lock is being held by more than one region, the count of the holders is shown.

## LOCK

Resource waited on or held.

### DATABASE

Database name.

### DCB/AREA

DCB number of a full function DL/I database data set or the area name for a Fast Path DEDB.

### F

Fast Path indicator:

blank	Non-Fast Path lock
B	B-lock for share level greater than 2, DS-BUSY, DATASET, and COMMAND
F	Fast Path lock
N	Non-updatable Fast Path area lock
P	P-lock for share level greater than 2, EXTEND, and BID

### G

Global indicator:

blank	Local lock
G	Global lock

## RBA/TYPE

In case of an RID (record lock) or BID (block lock), this field contains an identifier IMS uses to control access to the resource. The rules IMS uses to construct the ID are listed below to help in problem determination.

### FP DEDB

The ID is the high-order three bytes of the RBA of the VSAM control interval containing the segment.

### FP MSDB

The ID is the address of the segment prefix in the IMS control region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

### HDAM OSAM

The ID is the RBA of the block containing the segment.

#### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (X' 40' ) of the high order byte is turned on.

#### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

#### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

There are many types of locks to serialize various resources. The most common ones are block lock (BID) and record lock (RID). Block lock is used to serialize the update to a block or CI. Record lock is used to deny access to uncommitted data. In some cases, this service uses resource ID values that do not represent a relative byte address. In these cases, the following text is displayed in place of the actual ID:

#### COMMAND

Used for global commands, IDENTIFY to IRLM, or IRLM NOTIFY.

#### DATASET

Used at data set open for buffer invalidation notifies.

#### DS-BUSY

Data set busy. Used to ensure no CI or CA split during a read.

#### EXTEND

Used to serialize a data set extension.

The following locks are Fast Path only:

#### AREA

Used by Fast Path to lock an entire area. This is used mostly during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

#### DUMMY

Force release of all global locks.

#### NONDBRC

Database is not registered with DBRC.

#### WAIT CNT

Total number of waiters on the lock.

## WAITERS

Identifies the first two waiters for that locked resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS Selected Lock Pop-up

BMC SOFTWARE ----- IRLM LOCKS HELD BY RESRC ----- PERFORMANCE MGMT									
SERV ==> LHRES		INPUT		16: 57: 18		INTVL=> 3		LOG=> N TGT==> IMSA	
PARM ==>						LINE		1 OF 1 SCROLL=> CSR	
----- LOCK----- CUR WAIT									
DATABASE		DCB/AREA		RBA/TYPE		F G STA		HOLDERS CNT WAITERS	
-----									
CUSTHDAM		1		0000490E		P G UPD		002-I16XBMPW	
=====									
----- HOLDERS----- WAITERS----- REQ									
RGN T		STC/JOB		PSB		STATUS		RGN T STC/JOB PSB STAT	
-----									
002 B I16XBMPW PTEST02 WT-IRLM									
***** END OF DATA *****									

ACTV-DB2	Active in DB2 (Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on). The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	Region waiting for an AO message. Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch). <b>Note:</b> The block mover latch comprises several smaller latches. To find which BML latch a region is waiting for, use the LATCH service.
WAIT-DLI	Region waiting for DL/I.

WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCRB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.

WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDb segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### STC/JOB

Name of the job or started task. For an online IMS on a partner IRLM, this is the subsystem ID of that IMS. For an IMS internal task, this is the 3-byte field task ID (for example, XFP for Fast Path internal task).

#### T

Region type. Possible types:

B	Batch message processing (BMP) or Java batch message processing (JBP) region
D	DBCTL thread (CICS or ODBA)
F	Fast Path region (IFP)
L	Local batch IMS region
M	Message processing (MPP) or Java message processing (JMP) region
S	IMS internal task region
T	TPI (explicit APPC program)
X	External source (from a partner IRLM)

#### WAITERS

Lists all the waiters for the lock.

REQ STAT	Requested lock state:
EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

# LHUSR - IRLM Locks Held by User (IRLM 1.5)

BMC SOFTWARE ----- IRLM LOCK HELD BY USER ----- PERFORMANCE MGMT									
SERV ==> LHUSR		INPUT		14: 25: 00		INTVL=> 3		LOG=> N	
PARM ==>						LINE		1 OF 1	
<< EXPAND >>									
-----USER-----					-----HOLDING-----			-- WAITING FOR--	
RGN	T	STC/JOB	PSB	TRANCODE	RO	SHR	UPD	EXC	WAITERS
-----									
002	B	I 16XBMPW	PTEST02		0	0	1	0	1
***** END OF DATA *****									

**Description:** The LHUSR service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of the target IMS work units (dependent regions) that are holding and/or waiting for a lock. The regions are displayed in sequence by region number.

The user causing a lock contention is highlighted.

This service displays a superset of the regions shown in LCUSR. LCUSR lists only contentions, but LHUSR also includes any target IMS regions holding lock(s) with no one waiting.

**Select Code:** LHUSR

**Parameter:** None

**Expand:** You can select a detailed display (LUSRD) of any listed region, which identifies the resource(s) being waited for or held by that region. Position the cursor on the line for that region and press ENTER as indicated on the display by <<EXPAND>>. Pressing the PF3 (END) key from that display returns to this list.

**Scrolling:** The list of regions is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

## HOLDING

The locks being held by this region:

EXC	Number of EXCLUSIVE level locks being held
RO	Number of READ-ONLY level locks being held
SHR	Number of SHARE level locks being held
UPD	Number of UPDATE level locks being held

## USER

Identifies the target online IMS work units.

PSB	Name of the PSB (program specification block) being processed in this online region.
RGN	Region (PST) number.
STC/JOB	Name of the job or started task.

T	Region type, which can be one of the following:
B	Batch message processing (BMP) or Java batch message processing (JBP) region
D	DBCTL thread (CICS or ODBA)
F	Fast Path region (IFP)
M	Message processing (MPP) or Java message processing (JMP) region
T	TPI (explicit APPC program)
TRANCODE	The transaction code being processed by this online region, if the application is message driven.

#### WAITERS

The number of other regions waiting for one or more locks held by this region.

#### WAITING FOR

The region that this region is waiting for. There can be more than one region waiting, but only one is listed here.

RGN	Region (PST) number. If the region is not an online IMS region, this field is blank. If the region belongs to another IMS (different SSID), the region number is prefixed by an asterisk (*); for example, *001.
STC/JOB	The name of the job or started task. For an online IMS on a partner IRLM, this is the subsystem ID of that IMS. For an IMS internal process, this is the 3-byte IMS internal task field name; for example, XFP for Fast Path internal task.
T	Region type, which can be one of the following:
B	Batch message processing (BMP) or Java batch message processing (JBP) region
D	DBCTL thread (CICS or ODBA)
F	Fast Path region (IFP)
L	Local batch IMS region
M	Message processing (MPP) or Java message processing (JMP) region
S	IMS internal task
T	TPI (explicit APPC program)
X	External source (from a partner IRLM)

## LUSRD - IRLM Lock User Detail (IRLM 1.5)

```
BMC SOFTWARE ----- IRLM LOCK USER DETAIL ----- PERFORMANCE MGMT
SERV ==> LUSRD          INPUT    14: 15: 02  INTVL=> 3  LOG=> N  TGT==> IMSA
PARM ==> 001                      LINE    1 OF    1  SCROLL=> CSR
                                   << EXPAND >>

   RGN: 001  TYPE: BMP      STC: I16XBMPW   PSB: PTEST01   TRAN:
  IMSID: X16H          STATUS: ACTV-USR     PGM: PIWAIT    LTERM:

-----LOCK----- -STATE-  --COUNT--
DATABASE DCB/AREA RBA/TYPE F G  CUR REQ  HOLD WAIT
--- WAITING FOR-----
CUSTHDAM          1 00001000      SHR EXC    2    1    004-IMSMSG04  005-IMSMSG05
-----HOLDING-----
DEDB0001  AREA0001          F G  SHR SHR    10    0
***** END OF DATA *****
```

**Description:** The LUSRD service locates the partner IRLM of the target IMS. It then locates, identifies, and displays the locks the selected online IMS dependent region is holding or waiting for.

This display can be selected directly from the list of regions displayed by the LCUSR or LHUSR service by positioning the cursor on the line for the region and pressing ENTER.

**Select Code:** LUSRD

**Parameter:** Region number of a dependent region of the target online IMS.

**Expand:** The LUSRD display can be EXPANDED to the DLIST display for this region, which shows the current DL/I call in detail, including call function, PCB, and SSAs. Position the cursor on any LUSRD display line (line 5 or greater) and press ENTER.

**Scrolling:** The list of locks in the HOLDING area of the display is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

### Area 1

```
RGN: 001  TYPE: BMP   STC: I16XBMPW  PSB: PTEST01  TRAN:
IMSID: X16H          STATUS: ACTV-USR  PGM: PIWAIT   LTERM:
```

This area shows processing statistics about the selected region. If there is no lock activity, LUSRD shows this information only. The descriptions are arranged in alphabetical order.

**IMSID** The IMS subsystem ID.

**LTERM** If the application program is message driven and if the message originated from a terminal, this field shows the LTERM name of the terminal that submitted the transaction.

**PGM** The application program being executed for the PSB in the region. This can be different from the PSB if the region is a BMP.

PSB	The program specification block being processed in this region.
RGN	The online region (PST) number.
STATUS	Region status:
ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on).
	The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	Region waiting for an AO message.
	Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.

WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch).
	<b>Note:</b> The block mover latch comprises several smaller latches. To find which BML latch a region is waiting for, use the LATCH service.
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.

WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

STC (JOB) The online region started task name (or job name).

TRAN If the application program is message driven, this field shows the transaction code being processed by the region.

TYPE Region type:

BMP	Batch message processing region
BMP-IMP	Batch message processing region currently executing an implicit APPC/IMS transaction
BMP-OTM	Batch message processing region currently executing an OTMA transaction
BMP-WFI	Wait-for-input BMP
DBT	DBCTL CICS thread
FPU	Fast Path utility region
JBP	Java batch message processing region
JMP	Java message processing region

JMP-IMP	Java message processing region currently executing an implicit APPC/IMS transaction
JMP-OTM	Java message processing region currently executing an OTMA transaction
JMP-WFI	Wait-for-input JMP
MDP	Message-driven Fast Path region
MPP	Message processing region
MPP-IMP	Message processing region currently executing an implicit APPC/IMS transaction
MPP-OTM	Message processing region currently executing an OTMA transaction
MPP-WFI	Wait-for-input MPP
NDP	Non-message-driven Fast Path region
ODB	DBCTL ODBA thread
TPI	Message processing region currently executing an explicit CPI-C program

## Area 2

```

----- LOCK----- -STATE- -- COUNT--
DATABASE DCB/AREA RBA/TYPE F G CUR REQ HOLD WAIT
----- WAITING FOR( 34.6 MINUTES)-----
CUSTHDAM 1 00001000 SHR EXC 2 3* 004-IMSMSG04 005-IMSMSG05

```

This area shows locked resources that this region is waiting for, lock state, a total count of the lock's concurrent holders or waiters, and the holders of the locked resources. The descriptions are arranged in alphabetical order.

## COUNT

The total number of concurrent holders or waiters for the lock.

**HOLD** The total number of concurrent holders of the lock.

**WAIT** The total number of concurrent waiters on the lock.

## LOCK

Resource waited on or held.

### DATABASE

Database name.

### DCB/AREA

DCB number of a full function DL/I database data set or the area name for a Fast Path DEDB.

### F

Fast Path indicator:

blank	Non-Fast Path lock
B	B-lock for share level greater than 2, DS-BUSY, DATASET, and COMMAND
F	Fast Path lock
N	Non-updatable Fast Path Area lock

P P-lock for share level greater than 2, EXTEND, and BID

G

Global indicator:

blank	Local lock
G	Global lock

#### RBA/TYPE

In case of an RID (record lock) or BID (block lock), this field contains an identifier IMS uses to control access to the resource. The rules IMS uses to construct the ID are listed below to help in problem determination.

#### FP DEDB

The ID is the high-order three bytes of the RBA of the VSAM control interval containing the segment.

#### FP MSDB

The ID is the address of the segment prefix in the IMS control region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

#### HDAM OSAM

The ID is the RBA of the block containing the segment.

#### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (X' 40' ) of the high order byte is turned on.

#### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

#### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

There are many types of locks to serialize various resources. The most common ones are block lock (BID) and record lock (RID). Block lock is used to serialize the update to a block or CI. Record lock is used to deny access to uncommitted data. In some cases, this service uses resource ID values that do not represent a relative byte address. In these cases, the following text is displayed in place of the actual ID:

#### COMMAND

Used for global commands, IDENTIFY to IRLM, or IRLM NOTIFY.

#### DATASET

Used at data set open for buffer invalidation notifies.

#### DS-BUSY

Data set busy. Used to ensure no CI or CA split during a read.

#### EXTEND

Used to serialize a data set extension.

The following locks are Fast Path only:

#### AREA

Used by Fast Path to lock an entire area. This is used mostly during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

#### DUMMY

Force release of all global locks.

#### NONDBRC

Database is not registered with DBRC.

#### STATE

State of the lock.

CUR      Current state of the lock.

REQ      State of the lock requested by this region, which can be:

EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

#### HOLDERS

Identifies the first two holders of the lock on a resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.

## WAITING FOR

Identifies the locked resource (see LOCK description) that this region is waiting for, the lock state (see STATE description), and a count (see COUNT description) of the holders of that resource. If there is a lock, it is highlighted. If the region is not waiting for a lock, this area of the display is not shown.

<b><u>Area 3</u></b>									
----- LOCK----- - STATE- -- COUNT--									
DATABASE DCB/AREA RBA/TYPE F G CUR REQ HOLD WAIT									
:									
----- HOLDING----- - LONGEST WAITER-- - MI NUTES--									
CUSTHDAM 1 00002000 F G SHR SHR 10 1 003-IMSMSG03 34.5									
***** END OF DATA *****									

This area shows the locked resources held by a region, the lock state, a total count of the lock's concurrent holders or waiters, and the waiters for those locked resources. The descriptions are arranged in alphabetical order.

## HOLDING

Shows a scrollable list of the locked resources, as described for “Area 2” above (see LOCK, STATE, and COUNT), held by this region. The locked resources are displayed in alphabetical order.

## WAITERS

Identifies the first two waiters for that locked resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.

# IRLM - IRLM IMS Status (IRLM 2.1 and Later)

```
BMC SOFTWARE ----- IRLM IMS STATUS ----- PERFORMANCE MGMT
SERV ==> IRLM          INPUT  14:16:10 INTVL=> 3 LOG=> N TGT==> IMSA
PARM ==>                                LINE  1 OF  19 SCROLL=> CSR

SSID: IR21(001)  REL: 210  IRLM CONTENTION          IMS WAIT COUNT..... 0
PC(XMEM)..... NO      LOCAL..... 0
SCOPE..... LOCAL      GLOBAL..... 0  REQUEST COUNTERS:
GROUP..... N/A        REAL..... 0    LOCAL LOCK..... 3
LOCK TBL..... N/A     FALSE SLM..... 0  GLOBAL LOCK..... 26
OSAM TBL..... N/A     FALSE IRLM..... 0  UNLOCK..... 24
VSAM TBL..... N/A
EXIT ROUTINES CALLED:  CHANGE..... 0
IRLM STATUS:          SUSPEND..... 80  SENT TO SLM..... 0
INTERNAL TRACE NOT ACITVE RESUME..... 80  SYNC NOTIFY.....
STATUS..... 0        ASYNC NOTIFY.....
NOTIFY..... 0        XCF NOTIFY.....
DEADLOCK..... 0      PURGE.....
QUERY.....
TAKEOVER.....

IMS STATUS:          TYPE:
NORMAL              UPDATE
                   SHARE
***** END OF DISPLAY *****
```

**Description:** The IRLM service shows statistics and status information from the IRLM to which the target IMS is identified. Request counters, IMS wait counts, and status are for the target IMS only. IRLM contention and exit routine counts are for all IMSs using this IRLM.

**Select Code:** IRLM

**Parameter:** None

**Scrolling:** This display is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

```
Area 1

SSID: IR21(001)  REL: 210
PC(XMEM)..... NO
SCOPE..... LOCAL
GROUP..... N/A
LOCK TBL..... N/A
OSAM TBL..... N/A
VSAM TBL..... N/A
```

This area shows IRLM statistics. The descriptions are arranged in alphabetical order.

**GROUP**  
Name of the data sharing group this IRLM is a member of.

**LOCK TBL**  
Name of the XES lock structure used by this data sharing group.

OSAM TBL

Name of the OSAM cache structure used by this data sharing group.

PC(XMEM)

YES or NO (specified in DXRJPROC) defines whether IRLM uses the cross-memory program call (PC) service.

REL:

IRLM release number (for example, 210).

SCOPE

Specified in DXRJPROC.

LOCAL

Indicates that IRLM can be used only as the local lock manager (all sharing subsystems must use the same IRLM on the same OS/390).

GLOBAL

Indicates that the IRLM can participate with other IRLMs in an intraprocessor or interprocessor environment.

SSID:

IRLM subsystem name and ID (specified in the IRLM startup procedure DXRJPROC with the parameters IRLNM and IRLMID).

VSAM TBL

Name of the VSAM cache structure used by this data sharing group.

**Area 2**

```
IRLM CONTENTION
LOCAL.....0
GLOBAL.....0
REAL.....0
FALSE SLM.....0
FALSE IRLM.....0
```

The fields in this area show the amount and type of contention the IMSs using this IRLM have experienced since this IRLM was started. The contention counts are based solely on the number of suspended or rejected lock requests that originated from IMS subsystems using this IRLM. Suspended requests made to other IRLMs in the data sharing group are not included in these numbers, even when the cause of the contention may be a holder of a resource using this IRLM.

The descriptions are arranged in alphabetical order.

GLOBAL

A request to this IRLM for a global resource. A global resource is a shared level 2 or 3 database.

FALSE IRLM

Number of times a global request to this IRLM was granted by IRLM after SLM indicated contention. This is the longest code path length to grant a global request.

#### FALSE SLM

Number of times a global request to this IRLM was granted by SLM after the lock table indicated contention. If the lock table is too small, many resources share the same entry in the lock table.

#### REAL

Number of times a request to this IRLM for a global resource was suspended or rejected because of contention with another request to this IRLM or another IRLM in the IMSPLEX.

#### LOCAL

Number of times a request to this IRLM for a local resource was suspended or rejected because of contention with another request to this IRLM.

#### Area 3

IMS WAIT COUNT. . . . . 0

#### REQUEST COUNTERS:

LOCAL LOCK. . . . . 3

GLOBAL LOCK. . . . . 26

UNLOCK. . . . . 24

CHANGE. . . . . 0

SENT TO SLM . . . . . 0

SYNC NOTIFY. . . . . 1

ASYNCTIFY. . . . . 0

XCF NOTIFY. . . . . 0

VERIFY. . . . . 0

PURGE. . . . . 0

QUERY. . . . . 0

TAKEOVER. . . . . 0

This area shows the number of requests from the target IMS to this IRLM. The first field is:

#### IMS WAIT COUNT

Number of IMS regions or threads in the target IMS waiting for an IRLM request to be granted.

The following descriptions for REQUEST COUNTERS are in alphabetical order.

#### ASYNCTIFY

Number of asynchronous notifies from this subsystem to other subsystems. Database buffer invalidate, data set extension, and I/O errors are examples of when asynchronous notifies are sent.

#### CHANGE

Number of lock state change requests by this subsystem.

#### GLOBAL LOCK

Number of requests for global resource locks.

#### LOCAL LOCK

Number of requests for local resource locks.

**PURGE**

Number of times this subsystem requested IRLM to purge its locks or responded to an IRLM status exit.

**QUERY**

Number of query requests by this subsystem.

**SENT TO SLM**

Number of lock, unlock, and change requests sent to SLM.

**SYNC NOTIFY**

Number of synchronous notify requests by this subsystem.

**TAKEOVER**

Number of XRF backup requests to take over processing.

**UNLOCK**

Number of resource unlock requests by this subsystem.

**VERIFY**

Number of times this subsystem requested IRLM to verify the status of other subsystems with which it shares databases.

**XCF NOTIFY**

Number of notifies that required XCF (cross system coupling facility) communication.

**Area 4**

IRLM STATUS:  
INTERNAL TRACE NOT ACTIVE

This area shows the status of this IRLM. The descriptions are arranged in alphabetical order.

The following status messages can be displayed:

**DISCONNECTED FROM GROUP**

This IRLM is disconnected from the global sharing group.

**GLOBAL CONNECTION COMPLT**

Connection to global sharing environment completed.

**GLOBAL INIT IN PROGRESS**

Initialization of global sharing environment started.

**GLOBAL TERM IN PROGRESS**

Termination of global sharing environment started.

**INTERNAL TRACE (NOT) ACTIVE**

Internal trace is (not) active. This status always appears.

#### **Area 5**

```
EXIT ROUTINES CALLED:
SUSPEND. .... 0
RESUME. .... 0
STATUS. .... 0
NOTIFY. .... 0
DEADLOCK. .... 0
```

This area shows the number of times an exit routine is called by this IRLM for any IMS. The descriptions are arranged in alphabetical order.

#### **DEADLOCK**

This exit is entered by IRLM during a deadlock detection cycle if an IMS is involved in a deadlock. DFSDLKX0 calculates a worth value for each of its work units involved in the deadlock. IRLM then uses the value to select a candidate to break the deadlock.

#### **NOTIFY**

IRLM drives the notify exit to provide notification of certain events. For IMS, it calls DFSNOTX0 to handle these events. An example of an event requiring notification is data set extension by another IMS.

#### **RESUME**

IRLM calls the resume exit when a work unit can access the lock it is waiting for. For IMS, it calls DFSRESX0 to IPOST the work unit (ITASK) out of IWAIT.

#### **STATUS**

IRLM drives the status exit when it is told about a communication failure or a system failure of the partner IRLM.

#### **SUSPEND**

IRLM calls the suspend exit when a work unit must wait for a lock. For IMS, it calls DFSSUSX0 to IWAIT the work unit (ITASK).

#### **Area 6**

```
IMS STATUS:
NORMAL
```

This area shows the status of the target IMS in response to the activity of its partner IRLM. The descriptions are arranged in alphabetical order.

#### **NORMAL**

This IMS is in a normal state which can be qualified if any of the other status messages appear.

#### **ONLY ALLOW QUIT**

No other request for this IMS is accepted except quit.

#### **PURGE REQUIRED**

QUIT HELD=RETAIN was issued to keep all the locks held by this IMS. A purge is required to release the locks.

### QUIT ISSUED

A quit was issued for this IMS. Any work unit still in progress within this IMS is cancelled. All locks held by this IMS are released if **HELD=RELEASE** is issued or retained if **HELD=RETAIN** is issued. This information also is sent to the other IRLM in the data sharing group to update its ISL (Identified Subsystem List).

### SUBSYS LOCKS RETAINED

A **QUIT HELD=RETAIN** was issued for this IMS to stop all of its work in progress but to keep all its locks. This is usually an abnormal state (something is wrong with this IMS or its related IRLM).

#### Area 7

TYPE:  
UPDATE  
SHARE

This area displays the type of data sharing that this IMS does. The descriptions are arranged in alphabetical order.

#### NO SHARE

IMS does not share its databases with other IMS environments.

#### READ ONLY

IMS can only read its databases.

#### SHARE

IMS shares its databases with other IMS environments.

#### UPDATE

IMS can update its databases.

## IRLMG - IRLM GLOBAL STATUS (IRLM 2.1 and Later)

BMC SOFTWARE -----		IRLM GLOBAL STATUS -----		PERFORMANCE MGMT	
SERV ==> IRLMG		INPUT 14:16:17 INTVL=> 3		LOG=> N TGT==> IMSA	
PARM ==>		LINE 1 OF 31		SCROLL=> CSR	
SSID: IR21(001) REL: 210		IRLM CONTENTION COUNTS:		LOCAL IMS REQUEST COUNTS:	
PC(XMEM).....YES		LOCAL.....0		LOCAL LOCK.....0	
SCOPE.....GLOBAL		GLOBAL.....0		GLOBAL LOCK.....2	
GROUP.....IMSNWAY		REAL.....0		UNLOCK.....0	
LOCK TBL...IMS_LOCKTAB01		FALSE SLM.....0		CHANGE.....0	
OSAM TBL.....IMS_OSAM01		FALSE IRLM.....0		SENT TO SLM.....2	
VSAM TBL.....IMS_VSAM01				SYNC NOTIFY.....0	
		EXIT ROUTINES CALLED:		ASYNC NOTIFY.....0	
IRLM STATUS:		SUSPEND.....2		XCF NOTIFY.....0	
GLOBAL CONNECTION CMLPT		RESUME.....2		VERIFY.....0	
INTERNAL TRACE NOT ACTIVE		STATUS.....0		PURGE.....0	
		NOTIFY.....0		QUERY.....0	
		DEADLOCK.....0		TAKEOVER.....0	
-----SYSPLEX XES STATISTICS-----					
DEADLOCK PARMS... (5S, 1)		CURRENT XES REQUESTS:		CURRENT LOCK STRUCTURE:	
LOCAL.....0		TOTAL.....8		REQUESTS/SLOT.....0	
GLOBAL.....0		CONTENTION DELAY...0%		SLOTS USED.....0	
MAXCSA PARM...3,072,000		FALSE CONT. DELAY...0%		RECORD AREA USAGE....0%	
CURRENT.....183,296		SERVICE DELAY.....0%			
HIGHEST.....183,296					
===== IDENTIFIED SUBSYSTEMS =====					
SUBSYSID	STATE	TYPE	INTENT	OTHER STATUS	IRLM ID
-----					
X19H	ACTIVE	ONLINE	UPDATE	NORMAL	001
	PRIMARY	LOCAL	SHARE		
Y19H	ACTIVE	REMOTE	UPDATE	NORMAL	002
	PRIMARY	LOCAL	SHARE		
***** END OF DISPLAY *****					

**Description:** The IRLMG service shows statistics and status information from the IRLM to which the target IMS is identified. It includes information related to all subsystems using this IRLM. It also shows the status of all IMS subsystems in the IMSPLEX (N-way sharing group) and which IRLM they use.

**Select Code:** IRLMG

**Parameter:** None

**Scrolling:** This display is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

**Area 1**

```
SSID: IR21(001)  REL: 210
PC(XMEM)..... YES
SCOPE..... GLOBAL
GROUP..... IMSNWAY
LOCK TBL... IMS_LOCKTAB01
OSAM TBL..... IMS_OSAM01
VSAM TBL..... IMS_VSAM01
```

This area shows IRLM statistics. The descriptions are arranged in alphabetical order.

**GROUP**

Name of the data sharing group this IRLM is a member of.

**LOCK TBL**

Name of the XES lock structure used by this data sharing group.

**OSAM TBL**

Name of the OSAM cache structure used by this data sharing group.

**PC(XMEM)**

YES or NO (specified in DXRJPROC) defines whether IRLM uses the cross-memory program call (PC) service.

**REL:**

IRLM release number (for example, 210).

**SCOPE**

Specified in DXRJPROC.

**LOCAL**

Indicates that IRLM can be used only as the local lock manager (all sharing subsystems must use the same IRLM on the same OS/390).

**GLOBAL**

Indicates that the IRLM can participate with other IRLMs in an intraprocessor or interprocessor environment.

**SSID:**

IRLM subsystem name and ID (specified in the IRLM startup procedure DXRJPROC with the parameters IRLNM and IRLMID).

**VSAM TBL**

Name of the VSAM cache structure used by this data sharing group.

## Area 2

```
IRLM CONTENTION COUNTS:
LOCAL. .... 0
GLOBAL. .... 0
REAL. .... 0
FALSE SLM .... 0
FALSE IRLM .... 0
```

The fields in this area show the amount and type of contention the IMSs using this IRLM have experienced since this IRLM was started. The contention counts are based solely on the number of suspended or rejected lock requests that originated from IMS subsystems using this IRLM. Suspended requests made to other IRLMs in the data sharing group are not included in these numbers, even when the cause of the contention may be a holder of a resource using this IRLM.

The descriptions are arranged in alphabetical order.

### GLOBAL

A request to this IRLM for a global resource. A global resource is a shared level 2 or 3 database.

### FALSE IRLM

Number of times a global request to this IRLM was granted by IRLM after SLM indicated contention. This is the longest code path length to grant a global request.

### FALSE SLM

Number of times a global request to this IRLM was granted by SLM after the lock table indicated contention. If the lock table is too small, many resources share the same entry in the lock table.

### REAL

Number of times a request to this IRLM for a global resource was suspended or rejected because of contention with another request to this IRLM or another IRLM in the IMSPLEX.

### LOCAL

Number of times a request to this IRLM for a local resource was suspended or rejected because of contention with another request to this IRLM.

### Area 3

```
LOCAL IMS REQUEST COUNTS:
LOCAL LOCK.....0
GLOBAL LOCK.....2
UNLOCK.....0
CHANGE.....0
SENT TO SLM.....2
SYNC NOTIFY.....0
ASYNC NOTIFY.....0
XCF NOTIFY.....0
VERIFY.....0
PURGE.....0
QUERY.....0
TAKEOVER.....0
```

This area shows the number of requests from any subsystem (IMS or CICS for example) to this IRLM.

The following descriptions for REQUEST COUNTS are in alphabetical order.

#### ASYNC NOTIFY

Number of asynchronous notifies from this subsystem to other subsystems. Database buffer invalidate, data set extension, and I/O errors are examples of when asynchronous notifies are sent.

#### CHANGE

Number of lock state change requests by this subsystem.

#### GLOBAL LOCK

Number of requests for global resource locks.

#### LOCAL LOCK

Number of requests for local resource locks.

#### PURGE

Number of times this subsystem requested IRLM to purge its locks or responded to an IRLM status exit.

#### QUERY

Number of query requests by this subsystem.

#### SENT TO SLM

Number of lock, unlock, and change requests sent to SLM.

#### SYNC NOTIFY

Number of synchronous notify requests by this subsystem.

#### TAKEOVER

Number of XRF backup requests to take over processing.

#### UNLOCK

Number of resource unlock requests by this subsystem.

#### VERIFY

Number of times this subsystem requested IRLM to verify the status of other subsystems with which it shares databases.

## XCF NOTIFY

Number of notifies that required XCF (cross system coupling facility) communication.

### Area 4

#### IRLM STATUS:

GLOBAL CONNECTION COMPLT  
INTERNAL TRACE NOT ACTIVE

This area shows the status of this IRLM. The descriptions are arranged in alphabetical order.

The following status messages can be displayed:

#### DISCONNECTED FROM GROUP

This IRLM is disconnected from the global sharing group.

#### GLOBAL CONNECTION COMPLT

Connection to global sharing environment completed.

#### GLOBAL INIT IN PROGRESS

Initialization of global sharing environment started.

#### GLOBAL TERM IN PROGRESS

Termination of global sharing environment started.

#### INTERNAL TRACE (NOT) ACTIVE

Internal trace is (not) active. This status always appears.

### Area 5

#### EXIT ROUTINES CALLED:

SUSPEND. .... 2  
RESUME. .... 2  
STATUS. .... 0  
NOTIFY. .... 0  
DEADLOCK. .... 0

This area shows the number of times an exit routine is called by this IRLM for any IMS. The descriptions are arranged in alphabetical order.

#### DEADLOCK

This exit is entered by IRLM during a deadlock detection cycle if an IMS is involved in a deadlock. DFSDLKX0 calculates a worth value for each of its work units involved in the deadlock. IRLM then uses the value to select a candidate to break the deadlock.

#### NOTIFY

IRLM drives the notify exit to provide notification of certain events. For IMS, it calls DFSNOTX0 to handle these events. An example of an event requiring notification is data set extension by another IMS.

## RESUME

IRLM calls the resume exit when a work unit can access the lock it is waiting for. For IMS, it calls DFSRESX0 to IPOST the work unit (ITASK) out of IWAIT.

## STATUS

IRLM drives the status exit when it is told about a communication failure or a system failure of the partner IRLM.

## SUSPEND

IRLM calls the suspend exit when a work unit must wait for a lock. For IMS, it calls DFSSUSX0 to IWAIT the work unit (ITASK).

### **Area 6**

```
DEADLOCK PARMS. . . . (5S, 1)
  LOCAL. . . . . 0
  GLOBAL. . . . . 0
```

This area provides deadlock mode statistics. The descriptions are arranged in alphabetical order.

## DEADLOCK

The first number is the time in seconds between IRLM local deadlock cycles. The second number represents the number of local deadlock cycles before global deadlock detection is performed (defined in DEADLOCK of DXRJPROC).

## GLOBAL

Total number of global deadlocks.

## LOCAL

Total number of local deadlocks.

### **Area 7**

```
MAXCSA PARM . . . 3, 072, 000
  CURRENT. . . . . 183, 296
  HIGHEST. . . . . 183, 296
```

This area shows CSA/ECSA usage by IRLM. The descriptions are arranged in alphabetical order.

## CURRENT

Current CSA/ECSA usage.

## HIGHEST

CSA/ECSA usage high-water mark.

## MAXCSA PARM

Maximum amount of CSA/ECSA that the IRLM can use for its dynamic control blocks (specified in MAXCSA of DXRJPROC).

## Area 8

```
-----SYSPLEX XES STATISTICS-----
CURRENT XES REQUESTS:      CURRENT LOCK STRUCTURE:
TOTAL.....8      REQUESTS/SLOT.....0
CONTENTION DELAY...0%    SLOTS USED.....0
FALSE CONT. DELAY...0%    RECORD AREA USAGE....0%
SERVICE DELAY.....0%
```

This area shows information about current requests made to the XES locking structure by members of the IMS n-way sharing group. It also shows information about lock structure usage and capacity.

### CURRENT XES REQUESTS:

The following descriptions are arranged in alphabetical order:

#### CONTENTION DELAY

Percentage of current requests not granted immediately because of contention for the resource.

##### **Tuning Tip**

To reduce high contention delay, evaluate the application workload to determine if job scheduling can be altered to reduce contention between jobs. Application redesign may be necessary.

#### FALSE CONT. DELAY

Percentage of current requests not granted immediately because of false contention. This occurs when more than one resource hashes to the same lock table slot.

##### **Tuning Tip**

False contention causes increased locking overhead and should be minimized. Increasing the size of the lock structure allows for more slots, thus reducing false contentions.

#### SERVICE DELAY

Percentage of current requests not granted immediately because of internal XES service delays.

#### TOTAL

Total number of requests currently registered in the lock structure. This includes requests granted (lock is still held) or suspended (lock request not granted).

### CURRENT LOCK STRUCTURE:

The following descriptions are arranged in alphabetical order.

#### RECORD AREA USAGE

Percentage of available lock structure capacity currently in use by the sharing group. As the percentage in use reaches certain thresholds, data sharing capabilities are restricted. See IRLM message, DXR142I, in the *IMS Messages and Codes* manual provided by IBM.

## REQUESTS/SLOT

Ratio of current requests to number of lock table slots that are allocated.

## SLOTS USED

Number of lock table slots that currently have a resource mapped to them.

Area 9					
=====IDENTIFIED SUBSYSTEMS=====					
SUBSYSID	STATE	TYPE	INTENT	OTHER STATUS	IRLM ID
X19H	ACTIVE PRIMARY	ONLINE LOCAL	UPDATE SHARE	NORMAL NORMAL	001
Y19H	ACTIVE PRIMARY	REMOTE	UPDATE SHARE	NORMAL	002
*****END OF DISPLAY*****					

This area shows the subsystems that have identified themselves to IRLM. The description are arranged in alphabetical order.

## INTENT

The subsystem data sharing intent:

NO SHARE|SHARE

The subsystem either participates in data sharing or does not participate.

READ ONLY|UPDATE

The subsystem intent is update or read only.

## IRLM ID

ID of the IRLM to which this subsystem is identified. It is a unique value across the N-way sharing group.

## OTHER STATUS

Other subsystem status indicators:

NORMAL

Subsystem state is normal and none of the following occur.

PURGE ISSUED

Subsystem issued a PURGE to release all locks.

QUIT ISSUED

Subsystem issued a quit to stop all of its work in progress.

SUBSYS LOCKS RETAINED

A quit was issued with HELD=RETAIN to keep all of its locks on updated but uncommitted data.

## STATE

State of the subsystem:

ACTIVE|INACTIVE

Subsystem is active or inactive.

BACKUP|PRIMARY

Subsystem is the primary or the backup in the XRF complex. A non-XRF subsystem also is considered primary.

#### SUBSYSID

Subsystem name. For online IMS or DB2, this is the 4-byte subsystem ID. For IMS batch and CICS local DL/I, this is the jobname or started task name.

#### TYPE

Subsystem type:

##### BATCH

Subsystem is IMS batch (including CICS local DL/I).

##### DB2

Subsystem is DB2.

##### LOCAL

Subsystem is connected to this IRLM (local).

##### ONLINE

Subsystem is an IMS online system.

##### REMOTE

Subsystem is connected to its partner IRLM (IMS block level sharing only).

# LCRES - IRLM Lock Contention by Resource (IRLM 2.1)

BMC SOFTWARE ----- IRLM LOCK CONT BY RESRC ----- PERFORMANCE MGMT									
SERV ==>	LCRES	INPUT	16:56:54	INTVL=>	3	LOG=>	N	TGT==>	IMSA
PARM ==>				LINE	1	OF	1	SCROLL=>	CSR
EXPAND: LINESEL(DETAIL)									
-----LOCK----- CUR WAIT -----LONGEST WAIT-----									
DATABASE	DCB/AREA	RBA/TYPE	F G STA	HOLDERS	CNT	WAITER		MINUTES	
-----									
CUSTHDAM	1	0000615C	P G SHR	003-I16XBMPW	3	002-I16XBMPW		21.2	
***** END OF DATA *****									

**Description:** The LCRES service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of the locks related to the target IMS that are in lock contention (being waited for).

LCRES reports only lock wait conditions for requests that originated from an IMS identified to the same IRLM as the target IMS. If one of the local IMSs is waiting on a lock held by an external IMS, LCRES indicates that the lock is held externally, but does not provide any information about the holder.

**Select Code:** LCRES

**Parameter:** None

**Expand:** You can use the cursor to select a pop-up scrollable window, which lists the work units in contention for the selected lock.

**Scrolling:** The list of locks is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

## CUR STA

Current state of the lock. The state can be:

EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

## HOLDERS

Identifies the holder of the lock on a resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.

L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.
X - UNKNOWN	The holder is not known to this IRLM. The holder is identified to another IRLM (IRLM 2.1 or later).

If the lock is being held by more than one region, the count of the holders is shown.

## LOCK

Resource waited on or held.

### DATABASE

Database name.

### DCB/AREA

DCB number of a full function DL/I database data set or the area name for a Fast Path DEDB.

### F

Fast Path indicator:

blank	Non-Fast Path lock
B	B-lock for share level greater than 2, DS-BUSY, DATASET, and COMMAND
F	Fast Path lock
N	Non-updatable Fast Path area lock
P	P-lock for share level greater than 2, EXTEND, and BID

### G

Global indicator:

blank	Local lock
G	Global lock

## RBA/TYPE

In case of an RID (record lock) or BID (block lock), this field contains an identifier IMS uses to control access to the resource. The rules IMS uses to construct the ID are listed below to help in problem determination.

### FP DEDB

The ID is the high-order three bytes of the RBA of the VSAM control interval containing the segment.

### FP MSDB

The ID is the address of the segment prefix in the IMS control region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

#### HDAM OSAM

The ID is the RBA of the block containing the segment.

#### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (X' 40' ) of the high order byte is turned on.

#### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

#### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

There are many types of locks to serialize various resources. The most common ones are block lock (BID) and record lock (RID). Block lock is used to serialize the update to a block or CI. Record lock is used to deny access to uncommitted data. In some cases, this service uses resource ID values that do not represent a relative byte address. In these cases, the following text is displayed in place of the actual ID:

#### COMMAND

Used for global commands, IDENTIFY to IRLM, or IRLM NOTIFY.

#### DATASET

Used at data set open for buffer invalidation notifies.

#### DS-BUSY

Data set busy. Used to ensure no CI or CA split during a read.

#### EXTEND

Used to serialize a data set extension.

The following locks are Fast Path only:

#### AREA

Used by Fast Path to lock an entire area. This is used mostly during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

#### DUMMY

Force release of all global locks.

#### NONDBRC

Database is not registered with DBRC.

## -LONGEST WAIT-

### WAITER

Identifies the waiter waiting the longest number of minutes for a suspended lock on this resource. This does not include waiters in IMS subsystems that are identified to another IRLM. The waiter is identified as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.

### MINUTES

Number of minutes the lock request has been suspended.

### WAIT CNT

Total number of waiters on the lock. This does not include waiters in IMS subsystems identified to another IRLM.

### Selected Lock Pop-Up

BMC SOFTWARE				----- IRLM LOCK CONT BY RESRC -----				PERFORMANCE MGMT			
SERV ==> LCRES				INPUT 16: 56: 54 INTVL=> 3 LOG=>N TGT=> IMSA							
PARM ==>				LINE 1 OF 1				SCROLL=> CSR			
----- LOCK-----				CUR				WAIT ----- LONGEST WAIT-----			
DATABASE DCB/AREA				RBA/TYPE F G STA				HOLDERS CNT WAITER MINUTES			
CUSTHDAM				1 0000615C P G SHR				003-I16XBMPW 3 002-I16XBMPW 21. 2			
=====				=====				=====			
----- HOLDERS-----				----- WAITERS-----				REQ--- WAIT-			
RGN T STC/JOB PSB				STATUS				RGN T STC/JOB PSB STAT MINUTES			
003 B I16XBMPW				PTEST01 ACTV-USR				002 B I16XBMPW PTEST02 UPD 21. 2			
								004 B I16XBMPW PTEST04 UPD 1. 3			
*****				*****				*****			
				END OF DATA							

This pop-up shows the work units in contention for the selected lock. It is an EXPAND of a lock by cursor selection from the LCRES service.

## HOLDERS

Lists all the holders of the lock on IMSs identified to this IRLM.

### PSB

Program specification block being processed in the online IMS region. This is unknown when the holder is identified to another IRLM.

### RGN

Region (PST) number of the online IMS region; otherwise, the field is left blank. The region number is prefixed by an \* character if the region belongs to another IMS.

## STATUS

Region status:

**Note:** This is always unknown when the holder is an IMS identified to another IRLM.

ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on).  The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.
OPENING	Region in first CREATE THREAD process.

SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	Region waiting for an AO message.  Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch).  <b>Note:</b> The block mover latch comprises several smaller latches. To find which BML latch a region is waiting for, use the LATCH service.
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.

WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### STC/JOB

Name of the job or started task. This is unknown when the holder is identified to another IRLM. For an IMS internal task, this is the 3-byte field task ID (for example, XFP for Fast Path internal task).

T

Region type. Possible types:

B	Batch message processing (BMP) or Java batch message processing (JBP) region	
D	DBCTL thread (CICS or ODBA)	
F	Fast Path region (IFP)	
L	Local batch IMS region	
M	Message processing (MPP) or Java message processing (JMP) region	
S	IMS internal task region	
T	TPI (explicit APPC program)	
X	Holder is not known to this IRLM (holder is identified to another IRLM)	

#### WAITERS

Lists all the waiters for the lock. This does not include waiters in IMSs identified to another IRLM.

REQ STAT	Requested lock state:	
	EXC	Exclusive
	RO	Read-only
	SHR	Share
	UNK	Unknown (lock state in transition)
	UPD	Update

#### WAIT MINUTES

Number of minutes the lock request has been suspended.

# LCUSR - IRLM Lock Contention by User (IRLM 2.1)

BMC SOFTWARE ----- IRLM LOCK CONT BY USER ----- PERFORMANCE MGMT											
SERV ==> LCUSR		INPUT		14: 14: 54		INTVL=> 3		LOG>=N		TGT==> IMSA	
PARM ==>						LINE		1 OF		1 SCROLL=> CSR	
EXPAND: LI NESEL(LUSRD)											
-----USER-----				LOCKS		---WAITERS---		-----WAITING FOR-----			
RGN	T	STC/JOB	PSB	TRANCODE	HELD	TOTAL	LONGEST	RGN	T	STC/JOB	MINUTES
-----											
001	B	I 16XBMPW	PTEST01		10	03*	34.4				
002	B	I 16XBMPW	PTEST04		5	01	30.9	001	B	I 16XBMPW	34.4
003	B	I 16XBMPW	PTEST03		3	01	11.6	002	B	I 16XBMPW	30.9
004	B	I 16XBMPW	PTEST02		1	00		003	B	I 16XBMPW	11.6
***** END OF DATA *****											

**Description:** The LCUSR service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of the target IMS work units (dependent regions) that are involved in lock contentions (waiting or holding resources that someone is waiting for). The regions are displayed in sequence by region number.

**Color:** If you have a color monitor, the user causing a lock contention is highlighted in red.

**Select Code:** LCUSR

**Parameter:** None

**Expand:** You can select a detailed display (LUSRD) of any listed region, which identifies the resource(s) being waited for or held by that region. Position the cursor on the line for that region and press ENTER, as indicated on the display by EXPAND. Pressing the PF3 (END) key from that display returns to this list.

**Scrolling:** The list of regions is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

**LOCKS HELD**  
Total number of locks held by this user

**USER**  
Identifies the target online IMS work units that wait on resources locked by other users or held resources waited on by other users.

**PSB** Name of the PSB (program specification block) being processed in this online region.

**RGN** Region (PST) number.

**STC/JOB** Name of the job or started task.

T	Region type. Possible types:	
B	Batch message processing (BMP) or Java batch message processing (JBP) region	
D	DBCTL thread (CICS or ODBA)	
F	Fast Path region (IFP)	
M	Message processing (MPP) or Java message processing (JMP) region	
T	TPI (explicit APPC program)	
TRANCODE	Transaction code being processed by the online region, if the application is message driven.	

#### WAITERS

Number of other regions waiting for one or more locks held by this region and the longest time waiting:

TOTAL	Total waiters for resources held by this user. This does not include waiters identified to a different IRLM even when the other IRLM is part of the same N-way sharing group (the sharing of databases by more than two OS/390s). An * following this total indicates one or more of the total waiters is from a different IMS than the target IMS. These waiters from other IMSs, although included in this count, are not identified by this service.
LONGEST	Longest wait time of all local waiters for locks held by this user. There may be external waiters that have waited longer.

#### WAITING FOR

Region this region is waiting for. There can be more than one region waited for, but only one is listed here.

MINUTES	Time this user has been waiting for a lock.
RGN	Region (PST) number. This is unknown when the wait is for a holder identified to another IRLM. If the region is not an online IMS region, this field is blank. If the region belongs to another IMS (different subsystem ID), the region number is prefixed by an asterisk (for example, *001).
T	Region type. Possible types:  B     Batch message processing (BMP) or Java batch message processing (JBP) region D     DBCTL thread (CICS or ODBA) F     Fast Path region (IFP) L     Local batch IMS region M     Message processing (MPP) or Java message processing (JMP) region S     IMS internal task region T     TPI (explicit APPC program) X     Holder is not known to this IRLM (holder is identified to another IRLM)

## LHRES - IRLM Locks Held by Resources (IRLM 2.1)

BMC SOFTWARE			IRLM LOCKS HELD BY RESRC				PERFORMANCE MGM		
SERV ==>	LHRES	INPUT	16: 57: 09	INTVL=>	3	LOG=>	N	TGT==>	IMSA
PARM ==>				LINE	1	OF	6	SCROLL=>	CSR
EXPAND: LINESEL(DETAIL)									
-----LOCK-----			CUR		WAIT		-----LONGEST WAIT-----		
DATABASE	DCB/AREA	RBA/TYPE	F	G	STA	HOLDERS	CNT	WAITER	MINUTES
-----									
CUSTHDAM	1	0000615C	P	G	SHR	003-I 16XBMPW	3	002-I 16XBMPW	21. 2
		COMMAND	B	G	RO	S-DLI			
		COMMAND	F	G	RO	S-DLI			
CUSTHDAM	1	DATASET	B	G	RO	S-DLI			
CUSTHDAM	1	0000490E	P	G	UPD	002-I 16XBMPW			
CUSTHDAM	1	0000615C	P	G	SHR	003-I 16XBMPW	1	002-I 16XBMPW	
DBFSAMD3	CUSDB	AREA	N	G	RO	S-XFP			
***** END OF DATA *****									

**Description:** The LHRES service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of all the locks related to the target IMS that are being held (in contention or not).

**Select Code:** LHRES

**Parameter:** None

**Expand:** You can use the cursor to select a pop-up scrollable window, which lists all the work units in contention for the selected lock.

**Scrolling:** The list of locks is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

### CUR STA

Current state of the lock. The state can be:

EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

### HOLDERS

Identifies the holder of the lock on a resource as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.

S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.
X - UNKNOWN	The holder is not known to this IRLM. The holder is identified to another IRLM (IRLM 2.1 or later).
SHR CNT	If the lock is being held by more than one region, the count of the holders is shown.

## LOCK

Resource waited on or held.

### DATABASE

Database name.

### DCB/AREA

DCB number of a full function DL/I database data set or the area name for a Fast Path DEDB.

### F

Fast Path indicator:

blank	Non-Fast Path lock
B	B-lock for share level greater than 2, DS-BUSY, DATASET, and COMMAND
F	Fast Path lock
N	Non-updatable Fast Path Area lock
P	P-lock for share level greater than 2, EXTEND, and BID

### G

Global indicator:

blank	Local lock
G	Global lock

## RBA/TYPE

In case of an RID (record lock) or BID (block lock), this field contains an identifier IMS uses to control access to the resource. The rules IMS uses to construct the ID are listed below to help in problem determination.

### FP DEDB

The ID is the high-order three bytes of the RBA of the VSAM control interval containing the segment.

### FP MSDB

The ID is the address of the segment prefix in the IMS control region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

#### HDAM OSAM

The ID is the RBA of the block containing the segment.

#### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (X' 40' ) of the high order byte is turned on.

#### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

#### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

There are many types of locks to serialize various resources. The most common ones are block lock (BID) and record lock (RID). Block lock is used to serialize the update to a block or CI. Record lock is used to deny access to uncommitted data. In some cases, this service uses resource ID values that do not represent a relative byte address. In these cases, the following text is displayed in place of the actual ID:

#### COMMAND

Used for global commands, IDENTIFY to IRLM, or IRLM NOTIFY.

#### DATASET

Used at data set open for buffer invalidation notifies.

#### DS-BUSY

Data set busy. Used to ensure no CI or CA split during a read.

#### EXTEND

Used to serialize a data set extension.

The following locks are Fast Path only:

#### AREA

Used by Fast Path to lock an entire area. This is used mostly during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

#### DUMMY

Force release of all global locks.

#### NONDBRC

Database is not registered with DBRC.

#### -LONGEST WAIT-

Identifies the waiter waiting the longest number of minutes for a suspended lock on this resource. This does not include waiters in IMS subsystems that are identified to another IRLM. The waiter is identified as:

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
MINUTES	Number of minutes the lock request has been suspended.
WAIT CNT	Total number of waiters on the lock. This does not include waiters in IMS subsystems identified to another IRLM.

#### Selected Lock Pop-up

BMC SOFTWARE ----- IRLM LOCKS HELD BY RESRC ----- PERFORMANCE MGMT											
SERV ==> LHRES		INPUT		16: 57: 18		INTVL=> 3		LOG=> N		TGT==> IMSA	
PARM ==>						LINE		1 OF		1 SCROLL=> CSR	
----- LOCK----- CUR										WAIT ----- LONGEST WAIT-----	
DATABASE DCB/AREA		RBA/TYPE F G STA		HOLDERS		CNT		WAITER		MINUTES	
-----										-----	
CUSTHDAM		1 0000615C P G SHR		003- I16XBMPW		3		002- I16XBMPW		21. 2	
=====										=====	
----- HOLDERS-----										----- WAITERS----- REQ-- WAIT-	
RGN T STC/JOB		PSB		STATUS		RGN T STC/JOB		PSB		STAT MINUTES	
-----										-----	
003 B I16XBMPW		PTEST01		ACTV-USR		002 B I16XMPBW		PTEST02		UPD 21. 2	
						004 B I16XMPBW		PTEST04		UPD 1. 3	
*****										*****	
END OF DATA										*****	

This pop-up shows a scrollable list of all the work units in contention for the selected lock. It is an EXPAND of a lock by cursor selection from the LHRES service.

#### HOLDERS

Lists all the holders of the lock.

#### PSB

Program specification block being processed in the online IMS region. This is unknown when the holder is identified to another IRLM.

#### RGN

Region (PST) number of the online IMS region; otherwise, the field is left blank. The region number is prefixed by an \* character if the region belongs to another IMS.

## STATUS

Region status:

**Note:** This will always be unknown when the holder is an IMS identified to another IRLM.

ACTIVE	Active in a nonspecific process.
ACTV-BKO	Region in dynamic backout.
ACTV-DB2	Active in DB2 (Event Collector must be active).
ACTV-DBR	Region active in DBRC.
ACTV-DLI	Region active in DL/I.
ACTV-MQS	Region active in MQSeries (Event Collector must be active).
ACTV-SCH	Region active in nonspecific CREATE THREAD process.
ACTV-USR	Region active in the application program.
IDLE	Region waiting for non-WFI input to process.
IDLE-HOT	Pseudo-WFI region waiting for input from the same transaction.
IDLE-WFI	Region waiting for WFI or Fast Path BALG input to process.
INACTIVE	Region defined but not started (not yet signed on).  The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.

WAIT-AOI	Region waiting for an AO message.  Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	Region in scheduling (waiting for block mover latch).  <b>Note:</b> The block mover latch comprises several smaller latches. To find which BML latch a region is waiting for, use the LATCH service.
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.
WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.

WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

#### STC/JOB

Name of the job or started task. This is always unknown when the holder is identified to another IRLM. For an IMS internal task, this is the 3-byte field task ID (for example, XFP for Fast Path internal task).

T

Region type. Possible types:

B	Batch message processing (BMP) or Java batch message processing (JBP) region	
D	DBCTL thread (CICS or ODBA)	
F	Fast Path region (IFP)	
L	Local batch IMS region	
M	Message processing (MPP) or Java message processing (JMP) region	
S	IMS internal task region	
T	TPI (explicit APPC program)	
X	Holder is not known to this IRLM (holder is identified to another IRLM)	

#### WAITERS

Lists all the waiters for the lock. This does not include waiters in IMS subsystems identified to another IRLM.

REQ	STAT	Requested lock state:
	EXC	Exclusive
	RO	Read-only
	SHR	Share
	UNK	Unknown (lock state in transition)
	UPD	Update

#### WAIT MINUTES

Number of minutes the lock request has been suspended.

# LHUSR - IRLM Locks Held by User (IRLM 2.1)

BMC SOFTWARE ----- IRLM LOCK HELD BY USER ----- PERFORMANCE MGMT										
SERV ==> LHUSR		INPUT		14: 14: 54		INTVL=> 3		LOG=>N		TGT==> IMSA
PARM ==>				LINE			1 OF		1 SCROLL=> CSR	
EXPAND: LINESEL(LUSRD)										
-----USER-----				LOCKS		---WAITERS---		-----WAITING FOR-----		
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HELD TOTAL		LONGEST
RGN T		STC/JOB		PSB		TRANCODE		HE		

**Description:** The LHUSR service locates the partner IRLM of the target IMS. It then locates, identifies, and displays a list of the target IMS work units (dependent regions) that are holding and/or waiting for a lock. The regions are displayed in sequence by region number.

This service displays a superset of the regions shown in LCUSR. LCUSR lists only contentions, but LHUSR also includes any target IMS regions holding lock(s) with no regions waiting.

**Color:** If you have a color monitor, the user causing a lock contention is highlighted in red.

**Select Code:** LHUSR

**Parameter:** None

**Expand:** You can select a detailed display (LUSRD) of any listed region, which identifies the resource(s) being waited for or held by that region. Position the cursor on the line for that region and press ENTER as indicated on the display by EXPAND. Pressing the PF3 (END) key from that display returns to this list.

**Scrolling:** The list of regions is scrollable.

**Field Descriptions:** Each of the fields is shown and described below. The descriptions are arranged in alphabetical order.

**LOCKS HELD**  
Total number of locks held by this user

**USER**  
Identifies the target online IMS work units that wait on resources locked by other users or hold resources waited on by other users.

**PSB** Name of the PSB (program specification block) being processed in this online region.

**RGN** Region (PST) number.

**STC/JOB** Name of the job or started task.

T	Region type. Possible regions:	
B	Batch message processing (BMP) or Java batch message processing (JBP) region	
D	DBCTL thread (CICS or ODBA)	
F	Fast Path region (IFP)	
M	Message processing (MPP) or Java message processing (JMP) region	
T	TPI (explicit APPC program)	

TRANCODE Transaction code being processed by this online region, if the application is message driven.

#### WAITERS

Number of other regions waiting for one or more locks held by this region and the longest time waiting:

TOTAL Total waiters for resources held by this user. This does not include waiters identified to a different IRLM even when the other IRLM is part of the same N-way sharing group (the sharing of databases by more than two OS/390s).

An \* following this total indicates one or more of the total waiters is from a different IMS than the target IMS. These waiters from other IMSs, although included in this count, are not identified by this service.

LONGEST Longest wait time of all local waiters for locks held by this user. There may be external waiters that have waited longer.

#### WAITING FOR

Region this region is waiting for. There can be more than one region waited for, but only one is listed here.

MINUTES Time this user has been waiting for a lock.

RGN Region (PST) number. This is unknown when the wait is for a holder identified to another IRLM. If the region is not an online IMS region, this field is blank. If the region belongs to another IMS (different SSID), the region number is prefixed by an asterisk (\*); for example, \*001.

STC/JOB Name of the job or started task. This is unknown when the wait is for a holder identified to another IRLM. For an online IMS on a partner IRLM, this is the subsystem ID of that IMS. For an IMS internal process, this is the 3-byte IMS internal task field name; for example, XFP for Fast Path internal task.

T

Region type. Possible types:

- B Batch message processing (BMP) or Java batch message processing (JBP) region
- D DBCTL thread (CICS or ODBA)
- F Fast Path region (IFP)
- L Local batch IMS region
- M Message processing (MPP) or Java message processing (JMP) region
- S IMS internal task region
- T TPI (explicit APPC program)
- X Holder is not known to this IRLM (holder is identified to another IRLM)

## LUSRD - IRLM Lock User Detail (IRLM 2.1)

```
BMC SOFTWARE ----- IRLM LOCK USER DETAIL ----- PERFORMANCE MGMT
SERV ==> LUSRD          INPUT   14: 15: 02  INTVL=> 3  LOG=>N   TGT==> IMSA
PARAM ==> 001                      LINE   1 OF   1  SCROLL=> CSR
EXPAND: DLIST
      RGN: 001  TYPE: BMP      STC: I6XBMPW      PSB: PTEST01      TRAN:
      IMSID: X16H            STATUS: WT-IRLM      PGM: PI WAIT      LTERM:

-----LOCK----- - STATE-  --COUNT--
DATABASE DCB/AREA RBA/TYPE F G  CUR REQ  HOLD WAIT
-----WAITING FOR(   34.6 MINUTES)-----HOLDERS-----
CUSTHDAM          1 00001000      SHR EXC      2    3* 004-IMSMSG04 005-IMSMSG05
-----HOLDING-----LONGEST WAITER---MINUTES---
CUSTHDAM          1 00002000 F G  SHR SHR      10    1 003-IMSMSG03      34.5
***** END OF DATA *****
```

**Description:** The LUSRD service locates the partner IRLM of the target IMS. It then locates, identifies, and displays the locks the selected online IMS dependent region is holding or waiting for. It shows how long the selected region has waited for a lock and the wait time of the longest local waiter for each lock held by that region.

This display can be selected directly from the list of regions displayed by the LCUSR or LHUSR service by positioning the cursor on the line for the region and pressing ENTER.

**Color:** If you have a color monitor, a locked resource and the holders of the lock are highlighted in red.

**Select Code:** LUSRD

**Parameter:** Region number of a dependent region of the target online IMS.

**Expand:** The LUSRD display can be EXPANDED to the DLIST display for this region, which shows the current DL/I call in detail, including call function, PCB, and SSAs. Position the cursor on any LUSRD display line (line 5 or greater) and press ENTER.

**Scrolling:** The list of locks in the HOLDING area of the display is scrollable.

**Field Descriptions:** Each of the fields is shown and described below by display area.

**Area 1**

RGN: 001 TYPE: BMP STC: I16XBMPW PSB: PTEST01 TRAN:  
IMSID: X16H STATUS: ACTV-USR PGM: PIWAIT LTERM:

This area shows processing statistics about the selected region. If there is no lock activity, LUSRD shows this information only. The descriptions are arranged in alphabetical order.

IMSID IMS subsystem ID.

LTERM If the application program is message driven and if the message originated from a terminal, this field shows the LTERM name of the terminal that submitted the transaction.

PGM Application program being executed for the PSB in the region. This can be different from the PSB if the region is a BMP.

PSB Program specification block being processed in this region.

RGN Online region (PST) number.

STATUS Region status:

**Note:** This will always be unknown when the holder is an IMS identified to another IRLM.

ACTIVE Active in a nonspecific process.

ACTV-BKO Region in dynamic backout.

ACTV-DB2 Active in DB2 (Event Collector must be active).

ACTV-DBR Region active in DBRC.

ACTV-DLI Region active in DL/I.

ACTV-MQS Region active in MQSeries (Event Collector must be active).

ACTV-SCH Region active in nonspecific CREATE THREAD process.

ACTV-USR Region active in the application program.

IDLE Region waiting for non-WFI input to process.

IDLE-HOT Pseudo-WFI region waiting for input from the same transaction.

IDLE-WFI Region waiting for WFI or Fast Path BALG input to process.

INACTIVE	<p>Region defined but not started (not yet signed on).</p> <p>The region has initialized or started to initialize but has not completed signon or CREATE THREAD processing. For DBCTL, IMS preallocates threads (DBTs) up to the MINTHREAD specified value before they are actually needed. These DBTs will have an INACTIVE status. For other region types, this status should be displayed only briefly, until the region completes the first CREATE THREAD.</p>
OPENING	Region in first CREATE THREAD process.
SCH-BLR	Region in scheduling (active in block loader latch).
SCH-BLKM	Region in scheduling (active in block mover).
SCHDULE	Region in scheduling (CREATE THREAD).
TERMINAT	Region in region termination or abend.
WAIT-AOI	<p>Region waiting for an AO message.</p> <p>Await for AO occurs when a region issues a GMSG call with the wait option and DFSAOE00 has no message to return to it at that time.</p>
WAIT-BKO	Region currently in wait, but dynamic backout in progress.
WAIT-BLKM	Region waiting in block mover.
WAIT-BML	<p>Region in scheduling (waiting for block mover latch).</p> <p><b>Note:</b> The block mover latch comprises several smaller latches. To find which BML latch a region is waiting for, use the LATCH service.</p>
WAIT-DLI	Region waiting for DL/I.
WAIT-INT	Region in scheduling (waiting for database intent).
WAITING	Region in nonspecific wait.
WLT-xxxx	Region waiting for a latch with IMS latch ID of xxxx. For more information see “LATCH - Latch Summary” on page 272.
WT-CMDP	Region waiting for a pending /DBD or /DBR command to complete.
WT-DBRC	Region waiting for DBRC.
WT-DMBP	Region in scheduling (waiting for DMB pool).
WT-EPCB	Region in scheduling (waiting for EPCB pool).
WT-IRLM	Region waiting for IRLM.
WT-ISWCH	Region did ISWITCH and is waiting in IMS dispatcher.

WT-NTFY	Region waiting for asynchronous notify(s) to complete (IRLM must be active). An asynchronous notify could be buffer invalidation, for example.
WT-OSAM	Region waiting for OSAM.
WT-PI	Region waiting for a program isolation lock to be freed.
WT-PSBP	Region waiting for PSB pool.
WT-PSBW	Region waiting for PSBW pool.
WT-SCHD	Region waiting in a nonspecific CREATE THREAD process.
WT-VSAM	Region waiting in VSAM.
WTF-ADSC	Region waiting in Fast Path for ADSC directory latch (DEDB area data sets).
WTF-AREA	Region waiting in Fast Path for DEDB area lock.
WTF-DEDB	Region waiting in Fast Path for DEDB ownership.
WTF-DMSH	Region waiting in Fast Path for DMAC share latch (DEDB area).
WTF-DMSY	Region waiting in Fast Path for DMAC synchronization latch (DEDB area).
WTF-FBFR	Region waiting in Fast Path for FIX buffer (DEDB).
WTF-FCMD	Region waiting in Fast Path for FNCB (Fast Path command) latch.
WTF-MSDB	Region waiting in Fast Path for MSDB latch.
WTF-OBA	Region waiting in Fast Path for overflow buffer interlock.
WTF-OCL	Region waiting in Fast Path for open/close latch.
WTF-RSL	Region waiting in Fast Path for resource latch.
WTF-SEG	Region waiting in Fast Path for MSDB segment.
WTF-SYNC	Region waiting in Fast Path for synchronization latch.

**Note:** A # character in place of the - character in a status indicates activity resulted from an IMS data capture exit (user exit for data propagation and site requirements) instead of the application program. For example, if a data capture exit made a request that is waiting for a program isolation lock, the STATUS field shows WT#PI rather than WT-PI. For more information about data capture exits, see the IBM System Administration Guide publication.

The Event Collector must be active to obtain the data capture exit status when the exit issues a DB2 call.

STC (JOB) Online region started task name (or job name).

TRAN	If the application program is message driven, this field shows the transaction code being processed by the region.	
TYPE	Region type:	
	BMP	Batch message processing region
	BMP-IMP	Batch message processing region currently executing an implicit APPC/IMS transaction
	BMP-OTM	Batch message processing region currently executing an OTMA transaction
	BMP-WFI	Wait-for-input BMP
	DBT	DBCTL CICS thread
	FPU	Fast Path utility region
	JBP	Java batch message processing region
	JMP	Java message processing region
	JMP-IMP	Java message processing region currently executing an implicit APPC/IMS transaction
	JMP-OTM	Java message processing region currently executing an OTMA transaction
	JMP-WFI	Wait-for-input JMP
	MDP	Message-driven Fast Path region
	MPP	Message processing region
	MPP-IMP	Message processing region currently executing an implicit APPC/IMS transaction
	MPP-OTM	Message processing region currently executing an OTMA transaction
	MPP-WFI	Wait-for-input MPP
	NDP	Non-message-driven Fast Path region
	ODB	DBCTL ODBA thread
	TPI	Message processing region currently executing an explicit CPI-C program

Area 2									
-----LOCK-----				-STATE-		--COUNT--			
DATABASE	DCB/AREA	RBA/TYPE	F G	CUR REQ	HOLD	WAIT			
-----WAITING FOR( 34.6 MINUTES)-----				-----HOLDERS-----					
CUSTHDAM	1	00001000		SHR EXC	2	3*	004-IMSMSG04	005-IMSMSG05	

This area shows locked resources that this region is waiting for, lock state, a total count of the lock's concurrent holders or waiters, and the holders of the locked resources. The descriptions are arranged in alphabetical order.

## COUNT

Total number of concurrent holders or waiters for the lock.

- HOLD** Total number of concurrent holders of the lock. If any external holders exist, the total increases by one, regardless of how many external holders there may be.
- WAIT** Total number of concurrent waiters on the lock. This total does not include waiters identified to a different IRLM even when the other IRLM is part of the same N-way sharing group (the sharing of databases by more than two OS/390s). An \* following this total indicates one or more waiters of this total is from a different IMS subsystem than the target IMS.

## HOLDERS

Identifies the first two holders of the lock, highlighted in red, on a resource as:

- \*nnn - xxxxxxxx Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
- nnn - xxxxxxxx Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
- L - xxxxxxxx Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
- S - xxxxxxxx IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
- X - xxxxxxxx Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.
- X - UNKNOWN Holder is not known to this IRLM, because it is identified to another IRLM.

## LOCK

Resource waited on or held.

### DATABASE

Database name.

### DCB/AREA

DCB number of a full function DL/I database data set or the area name for a Fast Path DEDB.

## F

Fast Path indicator:

blank	Non-Fast Path lock
B	B-lock for share level greater than 2, DS-BUSY, DATASET, and COMMAND
F	Fast Path lock
N	Non-updatable Fast Path Area lock
P	P-lock for share level greater than 2, EXTEND, and BID

## G

Global indicator:

blank	Local lock
G	Global lock

## RBA/TYPE

In case of an RID (record lock) or BID (block lock), this field contains an identifier IMS uses to control access to the resource. The rules IMS uses to construct the ID are listed below to help in problem determination.

### FP DEDB

The ID is the high-order three bytes of the RBA of the VSAM control interval containing the segment.

### FP MSDB

The ID is the address of the segment prefix in the IMS control region's private virtual storage.

**Note:** This is not an RBA; it is an address that may change each time IMS is started.

### HDAM OSAM

The ID is the RBA of the block containing the segment.

### Other OSAM

The ID is the RBN of the block containing the segment. In addition, bit 1 (X' 40' ) of the high order byte is turned on.

### VSAM ESDS

The ID is the RBA of the VSAM control interval containing the segment.

### VSAM KSDS

If the key is four bytes or less, the ID is the actual key. Otherwise, the ID is the hashed key, using the algorithm in DFSHASH0.

There are many types of locks to serialize various resources. The most common ones are block lock (BID) and record lock (RID). Block lock is used to serialize the update to a block or CI. Record lock is used to deny access to uncommitted data. In some cases, this service uses resource ID values that do not represent a relative byte address. In these cases, the following text is displayed in place of the actual ID:

#### COMMAND

Used for global commands, IDENTIFY to IRLM, or IRLM NOTIFY.

#### DATASET

Used at data set open for buffer invalidation notifies.

#### DS-BUSY

Data set busy. Used to ensure no CI or CA split during a read.

#### EXTEND

Used to serialize a data set extension.

The following locks are Fast Path only:

#### AREA

Used by Fast Path to lock an entire area. This is used mostly during /STOP AREA processing and XRF takeovers.

#### BFROVFL

Buffer Overflow (Fast Path only). This is used when a region has to use the overflow buffers (OBA parameter). IMS allows only one region at a time to use the overflow buffers, so it uses program isolation to serialize the process.

#### DUMMY

Force release of all global locks.

#### NONDBRC

Database is not registered with DBRC.

#### STATE

State of the lock.

CUR      Current state of the lock.

REQ      State of the lock requested by this region, which can be:

EXC	Exclusive
RO	Read-only
SHR	Share
UNK	Unknown (lock state in transition)
UPD	Update

#### WAITING FOR

Time this region has been waiting for a lock held by another region. It also identifies the locked resource (see LOCK description) this region is waiting for, the lock state (see STATE description), and a count (see COUNT description) of the holders of that resource. If there is a lock, it is highlighted in red. If the region is not waiting for a lock, this area of the display is not shown.

### Area 3

```
-----LOCK----- -STATE- --COUNT--
DATABASE DCB/AREA RBA/TYPE F G CUR REQ HOLD WAIT
:
-----HOLDING-----LONGEST WAITER---MINUTES--
CUSTHDAM      1 00002000 F G SHR SHR      10      1      003-IMSMG03      34.5
*****
***** END OF DATA *****
```

This area shows the locked resources held by a region, the lock state, a total count of the lock's concurrent holders or waiters, and the waiters for those locked resources. The descriptions are arranged in alphabetical order.

#### HOLDING

Shows a scrollable list of the locked resources, as described for “Area 2” above (see LOCK, STATE, and COUNT), held by this region. The locked resources are displayed in alphabetical order.

#### LONGEST WAITER

Identifies the waiter waiting the longest time for that locked resource. The waiter is identified as:

**Note:** For IRLM 2.1 and later, MVIMS shows only waiters on IMSs identified to the same IRLM as the target IMS. This is because each IRLM in the sharing group does not have information about waiters using other IRLMs.

*nnn - xxxxxxxx	Online IMS region of another local IMS using the same IRLM, where to another IMS, nnn is the region number, and xxxxxxxx is the region name.
nnn - xxxxxxxx	Online IMS region of the target IMS, where nnn is the region number and xxxxxxxx is the region name.
L - xxxxxxxx	Local DL/I batch or CICS local DL/I job using the same IRLM, where xxxxxxxx is the name of the job or started task.
S - xxxxxxxx	IMS internal process, where xxxxxxxx is the internal task field name; for example, XFP for Fast Path internal task.
X - xxxxxxxx	Online or batch IMS on a partner IRLM, where xxxxxxxx is the subsystem ID of an online IMS or the name of the job or started task for a batch IMS.

#### MINUTES

Wait time of the oldest waiter for a lock held by the region. This does not include waiters identified to a different IRLM.



---

## Chapter 17. WORKLOAD ANALYZER Displays (Quick Reference)

The following table shows the service select code and parameters for the Workload Analyzer displays of terminal, transaction, and user status and the workload wait services. It also lists the page where you can find a complete description of each service.

Table 4. Workload Analyzer Display Service Select Codes

Service Select Code	Parameter	See
DWAIT	[ parm] { FORMAT=%   #} { FLOWCOMP=I C   I Q   OQ   OC   SC   AP   SP   COMP   ALL} { TRANS=ACTIVE   ALL }	Chapter 22, “Requesting Workload Wait Data Collection (MWAIT)” on page 433  DWAIT displays data collected by a corresponding MWAIT request. For a description of the keyword parameters that can be used with an MWAIT request, see Table 5 on page 436.  Chapter 23, “Requesting Workload Wait Data Display (DWAIT)” on page 443  Chapter 24, “DWAIT - Workload Wait Display” on page 445
ISTAT	[ LTERM  LT=name   I D=name   START=name]	Chapter 18, “ISTAT - Terminal Input Status Display” on page 395
OSTAT	[ , LTERM  LT=name   , I D=user i d   , START=name   , COUNT>n   , SEV1   SEV2   , NONZERO   NZ]	Chapter 19, “OSTAT - Terminal Output Status Display” on page 407

Table 4. Workload Analyzer Display Service Select Codes (continued)

Service Select Code	Parameter	See
TRANQ	[ , START=name   , TRAN=name   , SEV1   SEV2   , SSQ	Chapter 20, “TRANQ - Transaction Queue Status Display” on page 419
USER	[ , SORT   S0=cc , L=va l ue   , USERID=i d   , USER=user   , TRAN=trancode   , LTERM=l term   , NODE=i d   , NALLOC   , ALLOC   , DEADQ   , CONV   , HELD   , PRST   , STOP   , EXCL   , TST   , XON   , DYN   , STATI C   , RESP   , RESPINP=i d   , NZ   , QCT>nnn]	Chapter 21, “USER - User Status Summary” on page 427

## Chapter 18. ISTAT - Terminal Input Status Display

```

BMC SOFTWARE ----- TERMINAL INPUT STATUS ----- PERFORMANCE MGMT
SERV ==> ISTAT INPUT 14:40:23 INTVL=> 3 LOG=> N TGT==> IMSA
PARM ==> BOLD10

LTERM      NODE      LAST
NAME      USER-ID  LINE/PTERM  TRAN/DEST  TERMINAL STATUS
A1QAAI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QAA01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QBBI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QBB01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QCCI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QCC01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QDDI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QDD01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QEEI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QEE01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QFFI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QFF01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QGGI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QGG01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QHHI 1A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
A1QHH01A  NONE      NONE      NONE      TERMINAL IS NOT ALLOCATED
BOLD0    NONE      BOLD0      NONE      TERMINAL IS NOT CONNECTED
BOLD1    NONE      BOLD1      NONE      TERMINAL IS NOT CONNECTED
BOLD10   NONE      BOLD10     NONE      TERMINAL IS NOT CONNECTED

```

### Description:

The terminal input status display shows the input status of terminals in IMS. It can also show the number of transactions still in the input queue from any IMS terminal.

**Note:** For dynamic terminals created by the Extended Terminal Option (ETO):

ISTAT may not show the dynamic terminals immediately.

LTERMs are organized by a hashing algorithm. MVIMS and MAINVIEW AutoOPERATOR services present LTERMs and nodes alphabetically. To do this, they use an index table to sort them. The table is refreshed according to a specified refresh interval and only when LTERM/node displays are requested. The default refresh interval is 10 seconds. If the time since the last refresh is greater than 10 seconds, the table is updated. Service requests that occur prior to interval expiration do not show dynamic terminals created during the interval.

The refresh interval determines how soon dynamic terminals are displayed by ISTAT. Short refresh intervals allow them to be displayed sooner, but they may consume more resources because the table is updated more frequently. Longer refresh intervals may consume less resources, but they may delay the display of recently created dynamic terminals. The amount of resources consumed by sorting depends on the size of your site's terminal network. To change the refresh interval, use the CBSINTVL parameter in

BBPARM member BBIISP00. The new value takes effect when the BBI-SS PAS restarts.

By default, the index table uses 200K of BBI-SS PAS extended private area storage. You can use the CBSORTSZ in BBPARM member BBIISP00 to change this value. Each terminal entry requires 12 bytes (8 bytes for the terminal ID and 4 bytes for the address of either the CNT or the CLB control block).

**Select Code:** ISTAT

**Parameter:** The following parameters can be used with this service:

ID

Enter an 8-character user ID or a qualified user ID; for example, ISTAT displays only those terminals that have a matching user ID.

LTERM|LT

Enter an 8-character LTERM name or a qualified name. ISTAT displays only those terminals that have a matching name, for example LTERM=A\*.

START

Enter a starting LTERM to page through LTERMs starting with a particular LTERM, for example START=PAY+.

**Scrolling:** The START= field in the ISTAT display contains the name of the first terminal in the next list to be displayed. Use this field to scroll through the list either forward or backward. Entering A1QFFI1A puts that terminal name at the top of the next displayed list.

Pressing ENTER repetitively scrolls forward a page at a time. For example, if the START= field is BOLD10 as shown in the sample display above, pressing ENTER puts BOLD10 at the top of the next list of displayed terminal names.

To scroll to another place on the list, enter the name of a terminal in the START= field. The \* qualifier can be used to indicate a generic terminal name. For example, enter A\* to scroll to the beginning of the list.

**Field Descriptions:** Each of the display fields is described below. The descriptions are arranged in alphabetical order.

LAST TRAN/DEST

This is the last transaction input from this terminal or the preset destination.

LTERM NAME

The LTERM name of the terminal. The LTERMs are listed in alphabetical order.

NODE LINE/PTERM

This is the node name if the terminal is a VTAM device. If it is a BTAM device, it is the line and PTERM number.

#### TERMINAL STATUS

This the status of the LTERM for inbound traffic (see “Input LTERM Status” on page 398 below).

#### USER-ID

The user ID signed on to the terminal, if this level of security is implemented.

**Note:** The user ID displayed is the security user ID, not the ETO structure name (SPQB).

---

## Input LTERM Status

Terminals that are defined to IMS as input terminals can be in any one of a large number of states. The ISTAT service tests the input status of a terminal in a predefined sequence so that the most critical information is always displayed. If special conditions are not found, the message returned is:

TERMINAL IS AOK.

The following status messages are possible. They are arranged in alphabetical order for easy reference.

- **ASYNCR MSG IN PROCESS**

For an LU6 session, the error message FMH is being sent.

- **BIND RECEIVED/SCHEDULED**

IMS has received a request from the other LU6 half-session or IMS is attempting to establish a session.

- **BIND SENT**

IMS has sent a BIND on this LU6 session.

- **BIS RECEIVED/SCHEDULED**

IMS asked the other LU6 half-session to stop initiating new brackets and the partner has agreed to stop or vice versa.

- **BIS SENT**

IMS has informed the other LU6 half-session that IMS will stop initiating new brackets on this session.

- **CANCEL RECEIVED/SCHEDULED**

IMS has received a request from the other LU6 half-session partner to cancel the input in progress or vice versa.

- **/CHE DUMPQ QUIESCE ENTERED**

IMS is shutting down as the result of a CHE DUMPQ QUIESCE command.

- **/CHE PURGE IN PROGRESS**

IMS is shutting down as the result of a /CHE PURGE command.

- **/CHE FREEZE QUIESCE ENTERED**

IMS is shutting down as the result of a /CHE FREEZE QUIESCE command.

- **/CHE DUMP QUEUES IN PROGRESS**

IMS is shutting down as the result of a /CHE DUMPQ command.

- **/CHE PURGE QUIESCE ENTERED**

IMS is shutting down as the result of a /CHE PURGE QUIESCE command.

- **CLEAR SENT**

IMS has sent a CLEAR on this LU6 session.

- **CLOSE VTAM ACB PENDING**

The IMS ACB is being closed so no communications using VTAM is possible. The non-VTAM terminals should work normally.

- **CLSDST REQUESTED**

This terminal is requesting to terminate its session with IMS. This node will be disconnected.

- **CONTROL REGION IS ABENDING**

IMS is shutting down as a result of an ABEND.

- **CONVERSATION HELD**

The terminal has had a /HOLD command issued to suspend and save a conversation. Conversation will be resumed (/RELEASE command) or terminated (/EXIT command), sometime later.

- **DEQUE PURGE IN PROGRESS**

The terminal has had a /DEQUEUE LINE or NODE PURGE command issued for it. All output will be cancelled.

- **/DEQUEUE PURGE ISSUED LTERM**

The LTERM has had a /DEQUEUE LTERM PURGE command issued for it. All output will be cancelled.

- **DESTINATION NOT FOUND**

The last input from this terminal was not a valid message switch, transaction code, or command.

- **DISCONNECT ALL LINES**

IMS is shutting down all lines.

- **DO NOT SEND QUEUED MSG**

The line or node associated with this LTERM has been /PSTOPped.

- **DYNAMIC TERMINAL NOT ALLOCATED**

This terminal was created dynamically by ETO and is not allocated to a node. This can occur under several circumstances; for example:

- Output was sent to this LTERM and ETO created the LTERM and user structure but not the node. If a node cannot be or is not created and allocated to this LTERM, messages are categorized as dead letters.
- A user logged off the system, either intentionally or through a network failure, and the status was such that the user structure and LTERM could not be deleted. This can occur, for example, if there are messages still queued to the LTERM or the user has an active conversation.

- **ECHO CURRENT INPUT**  
IMS echoes back whatever is received from this terminal. A /TEST command causes IMS to echo everything from the terminal until a /END command is entered.
- **ERROR FOUND ON LAST INPUT SEG**  
The last input from this terminal resulted in an error message.
- **/EXCL ISSUED FOR TERMINAL**  
The terminal is in EXCLUSIVE mode. The terminal has been placed in EXCLUSIVE mode using the /EXCL command. The terminal remains in EXCLUSIVE mode until a /END is issued.
- **FATAL ERP**  
IMS has received or is scheduling an UNBIND to send on this LU6 ISC session.
- **/IDLE COMMAND**  
The terminal has had a /IDLE LINE x PTERM y command issued for it. All input and output is terminated. The system is in shutdown.
- **/IDLE WITHOUT SHUTDOWN**  
The terminal has had a /IDLE LINE x PTERM y NOSHUT command issued for it. All input and output is terminated. The terminal remains in IDLE until it is restarted.
- **IMS SHUTDOWN IN PROGRESS**  
IMS is shutting down.
- **IMS SHUTDOWN MSG BEING SENT**  
IMS is shutting down. A shutdown message has been sent to this terminal.
- **INBOUND RU DID NOT REQ DR1 RESP**  
The last input from this terminal did not request a DR1 response.
- **INBOUND RU REQUESTED DR2 RESP**  
The last input from this terminal requested a DR2 response.
- **INBOUND RU REQUESTED EXCP/DRX**  
The last input from this terminal requested exception DRx response only.
- **IN BRACKETS**  
This LU6 session is in the in-bracket state.
- **INPUT STOPPED ON THIS LINE**  
The terminal has had a /STOP LINE or NODE command issued against it. IMS will not accept input until a /START command is issued.
- **I/O ERROR OCCURRED ON QUEUE**  
Some of the messages queued to this terminal are unavailable for transmission because I/O errors have occurred on one or more of the queue data sets.

- **I/O POOL SHORT ON STORAGE**

IMS is low on pool space.

- **IWAIT FOR POST=SCHED SEND**

This LU6 terminal is in an IMS wait for the VTAM notification that a SEND has been initiated.

- **/LOCK ISSUED FOR TERMINAL**

The terminal is LOCKed. The remote terminal user has locked the terminal using the /LOCK NODE or PTERM command. IMS does not receive and send messages to a terminal that is LOCKed; however, IMS continues to queue output traffic to a terminal in this state.

- **/LOOPTEST IN PROGRESS**

The terminal has had a /LOOPTEST LINE x PTERM y command issued for it. The terminal is in an output write loop that continues until a /END command terminates the looptest mode.

- **LOOPTEST IS PENDING**

The terminal has had a /LOOPTEST LINE x PTERM y command issued for it. The terminal is in an output write loop that continues until a /END command terminates the looptest mode.

- **LOST TERM**

The IMS lost terminal exit was entered for this terminal because of the loss of communications with the terminal.

- **LTERM HAS PAGED MSG IN PROCESS**

This LTERM is looking at a multi-page output message. The user may get to the end of the message or next message by using the PA1 and PA2 keys.

- **LTERM LOCKED FROM /LOCK**

The LTERM is LOCKed. The remote terminal user has locked the terminal using the /LOCK LTERM command or the MTO has issued the /LOCK LTERM x. IMS does not receive and send messages to an LTERM that is LOCKed; however, IMS continues to queue output traffic to a terminal in this state.

- **MSC INVALID BLOCK**

An invalid block has been received from the MSC partner.

- **NSPE CLEANUP RU**

The IMS lost terminal exit was entered for this terminal to allow IMS to cleanup after a procedure error.

- **NSPE NOTIFY RU**

The IMS lost terminal exit was entered for this terminal to inform IMS that a procedure error has occurred between VTAM and the terminal.

- **NSPE PROCEDURE ERROR RU**

The IMS lost terminal exit was entered for this terminal to inform IMS that a procedure error has occurred between IMS and VTAM.

- **OUTPUT ONLY TERMINAL**  
This terminal is for output only. No input is possible from this device.
- **PARTNERS SEQ NO. ARE AVAILABLE**  
The sequence numbers used by the LU6 session partner are known to IMS session recovery is possible.
- **PROTECT FOR INPUT**  
No output will be sent to this terminal, because it is PROTECTED. Input is expected by IMS.
- **QRI REPOSE REQUIRED**  
IMS must notify the other half of this LU6 half-session that a response is queued.
- **QUEUE ERROR WITH SYMSG**  
For LU6 sessions, an I/O error on the IMS queues is the cause of the SYMSG FMH that was sent to the partner half-session.
- **QUEUING IS STOPPED**  
The terminal is stopped for queueing. No output will be queued for this device.
- **/QUIESCE ISSUED**  
The node associated with this LU6 session has been /QUIESCED by the MTO.
- **RECOVERY REQUEST ON THIS NODE**  
For IMS SLU type P sessions (LU 0), IMS is attempting to resync the session after protocol errors occurred.
- **REJECT IF NOT INQUIRY TRAN**  
Terminal has been SYSGENed as non-checking and, therefore, inquiry-only. Update transactions are not allowed from this terminal.
- **RESPONSE OWED FOR BID**  
IMS has received a BID from this terminal and must send a response back to it.
- **RESPONSE OWED FOR BIS**  
IMS has received a bracket initiation stopped (BIS) from this terminal and must send a response back to it.
- **RESPONSE OWED FOR CANCEL**  
IMS has received a CANCEL from this terminal and must send a response back to it.
- **RESPONSE OWED FOR CHASE**  
IMS has received a CHASE from this terminal and must send a response back to it.
- **RESPONSE OWED FOR DATA**  
IMS has received data from this terminal and must send a response back to it.

- **RESPONSE OWED FOR LUS**

IMS has received an LUS (LU status) from this terminal and must send a response back to it.

- **RESPONSE OWED FOR RTR**

IMS has received a ready to receive (RTR) from this terminal and must send a response back to it.

- **RESTART RECEIVED/SCHEDULED**

IMS is in the process of attempting to re-establish an LU6 session or the other half-session is attempting to restart the session with IMS.

- **RPLS CREATED AND INITIALIZED**

The SNA control blocks for this LU6 session have been acquired and set up for communication on this session.

- **SBI RECEIVED/SCHEDULED**

IMS has been asked by the other LU6 half-session partner to stop initiating new brackets or IMS has requested cessation of brackets.

- **SCHED/STACK LOGON**

IMS has been asked to issue an open destination to this terminal.

- **SCHEDULE CANCEL**

For LU6 sessions, a CANCEL FMH must be sent.

- **SDT RECEIVED/SCHEDULED**

The other LU6 half-session partner has given the go ahead to start data traffic on the session or vice versa.

- **SDT SENT**

IMS has requested that data traffic commence on this LU6 session.

- **SECONDARY HS FLAG**

IMS is the secondary LU on this LU6 session.

- **SEL DSP - NO DYNAMIC SAPS**

IMS is out of dynamic SAPs. The system has entered selective dispatching and only selective IMS communications work will be dispatched.

- **SIMLOGON REQUESTED**

The other LU6 half-session is requesting to logon to IMS.

- **SNA NODE SENT QUIESCE**

This terminal is requesting to terminate its session with IMS. This node will be disconnected.

- **SPECIAL IN-RCV-MODE FLAG**

The LU6 session is waiting for specific protocol input from the partner half-session.

- **/STA DC NOT PERFORMED**

The IMS ACB is not open so IMS can not communicate using VTAM. The non-VTAM terminals should work normally. If a /STA DC is done, the VTAM terminal should become available.

- **STSN SENT**

IMS has sent a STSN on this LU6 session.

- **STSN RECEIVED/SCHEDULED**

During the processing of traffic on this LU6 session, IMS has received a sequence number establishment request from the other half-session partner or vice versa.

- **SYSTEM NOT ACCEPTING LOGONS**

IMS is not currently accepting logons so LOGON and /OPN commands do not work.

- **SYSTEM SHUT QUIESCE MSG RGN**

IMS is shutting down; message regions are being quiesced.

- **SYSTEM SHUT STOP INPUT**

IMS is shutting down; input has been stopped.

- **SYSTEM SHUT TERM MSG RGNS**

IMS is shutting down; message regions are being terminated.

- **SYSTEM SHUTDOWN PENDING**

IMS is shutting down; no more input is allowed from the terminal.

- **TERMINAL IN BRACKETS**

This terminal is in the in-bracket state.

- **TERMINAL IN NOBB STATE**

The LU6 session has no outstanding begin bracket.

- **TERMINAL IN RECEIVE MODE**

IMS is in the process of receiving input on this LU6 session.

- **TERMINAL IN SEND MODE**

IMS is in the process of sending output on this LU6 session.

- **TERMINAL IS AOK**

The terminal has no detected status problems. It should be working normally.

- **TERMINAL IS AWAITING RESPONSE**

This terminal has sent input to IMS and is waiting for acknowledgement. The terminal is defined as FORCRESP or TRANRESP, or a Fast Path transaction was entered.

- **TERMINAL IS IN A CONVERSATION**

The terminal has entered a conversational transaction and is currently in the middle of that conversation.

- **TERMINAL IS INOPERABLE**

The associated physical terminal is inoperable. IMS has detected some problem with the terminal and marked it inoperable; however, IMS continues to queue output traffic to a terminal in this state.

- **TERMINAL IS NOT ALLOCATED**

The terminal is not allocated. This is a VTAM subpool LTERM that is subject to dynamic allocation and is not currently in session. The node and the user ID will both be set to NONE.

- **TERMINAL IS NOT CONNECTED**

The terminal is not connected to IMS. The user may log on a VTAM terminal using LOGON APPLID(IMS), or the MTO can issue a /OPN NODE xxxx, or if a dial-up terminal, the user may call in to IMS.

- **TERMINAL IS STOPPED FOR INPUT**

The terminal has had a /STOP LINE or NODE command issued against it. No input will be received by IMS until a /START command is issued.

- **TERMINAL IS WAITING FOR RESP**

This terminal has sent input to IMS for a response mode transaction. IMS has not responded and so the LTERM remains unavailable.

- **TERMINAL NOT SIGNED ON**

This terminal requires that the operator sign on before using it. The operator should do a /SIGN command.

- **/TEST ISSUED FOR TERMINAL**

The terminal is in TEST. The terminal has been placed in TEST using the /TEST command. The TERMINAL will remain in TEST until a /END is issued.

- **THIS LINE IS INACTIVE**

The line or node associated with this LTERM is inactive, because a session never started or there are no messages queued.

- **TPEND HALT CANCEL PENDING**

VTAM is being shut down so IMS cannot communicate using VTAM. The non-VTAM terminals should work normally.

- **TPEND HALT NON-QUICK PENDING**

VTAM is being shut down so IMS cannot communicate using VTAM. The non-VTAM terminals should work normally.

- **TPEND HALT QUICK PENDING**

VTAM is being shut down so IMS cannot communicate using VTAM. The non-VTAM terminals should work normally.

- **UNBIND RECEIVED**

This is a VTAM terminal that has sent an UNBIND to IMS. The terminal is disconnected from IMS. To get the terminal back, the MTO may issue a /OPN command or the terminal may log on again.

- **UNBIND RECEIVED/SCHEDULED**  
IMS has received or is scheduling an UNBIND to send on this LU6 half-session.
- **UNBIND SENT**  
This terminal is terminating its session with IMS. This node will be disconnected.
- **VTAM ACB NOT OPEN**  
The IMS ACB is not open so IMS can not communicate using VTAM. The non-VTAM terminals should work normally.
- **WAIT FOR LUS-ABORT**  
The LU6 session is in the process of being ABORTed. IMS is waiting for the LU ABORT status FMH.
- **WAIT FOR POOL SPACE - MFSP**  
This terminal is waiting for space to be freed in pool MFSP before IMS can continue servicing it.
- **WAIT FOR POOL SPACE - I/OP**  
This terminal is waiting for space to be freed in pool I/OP (CIOP) before IMS can continue servicing it.
- **WAITING FOR LOSTERM**  
IMS has lost communications with this VTAM terminal and is waiting for VTAM to invoke the lost terminal exit.
- **WE SPEAK FIRST**  
For this LU6 session, IMS is the primary half-session.

## Chapter 19. OSTAT - Terminal Output Status Display

```
BMC SOFTWARE ----- TERMINAL OUTPUT STATUS -----PERFORMANCE
MGMT SERV ==> OSTAT INPUT 14:40:23 INTVL=> 3 LOG=> N TGT==> IMSA PARM
==> BOLD1 IMS-VTAM INTERFACE NORMAL IMS STATUS NORMAL -LTERM-- -SIGNON-
NODE/SP TERM OUTPUT LAST --NAME-- ---ID--- LN#/PTRM TYPE MSG # TRN/DEST
TERMINAL STATUS----- A1QAA11A L1WA1I01 SUBPL 0 TERMINAL
IS NOT ALLOCATED A1QAA01A L1WA1O01 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QBB11A L1WA1I02 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QBB01A L1WA1O02 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QCC11A L1WA1I03 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QCC01A L1WA1O03 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QDD11A L1WA1I04 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QDD01A L1WA1O04 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QEE11A L1WA1I05 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QEE01A L1WA1O05 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QFF11A L1WA1I06 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QFF01A L1WA1O06 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QGG11A L1WA1I07 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QGG01A L1WA1O07 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QHH11A L1WA1I08 SUBPL 0 TERMINAL IS NOT ALLOCATED
A1QHH01A L1WA1O08 SUBPL 0 TERMINAL IS NOT ALLOCATED
BOLD0 BOLD0 3270V 0 TERMINAL IS INOPERABLE
```

### Description:

The terminal output status display shows the status of terminals in IMS with outbound messages. It can also show the number of messages still in the output queue destined for an IMS terminal.

**Note:** For dynamic terminals created by the Extended Terminal Option (ETO):

OSTAT may not show the dynamic terminals immediately.

LTERMs are organized by a hashing algorithm. MVIMS and MAINVIEW AutoOPERATOR services present LTERMs and nodes alphabetically. To do this, they use an index table to sort them. The table is refreshed according to a specified refresh interval and only when LTERM/node displays are requested. The default refresh interval is 10 seconds. If the time since the last refresh is greater than 10 seconds, the table is updated. Service requests that occur prior to interval expiration do not show dynamic terminals created during the interval.

The refresh interval determines how soon dynamic terminals are displayed by OSTAT. Short refresh intervals allow them to be displayed sooner, but they may consume more resources because the table is updated more frequently. Longer refresh intervals may consume less resources, but they may delay the display of recently created dynamic terminals. The amount of resources consumed by sorting depends on the size of your site's terminal network. To

change the refresh interval, use the CBSINTVL parameter in BBPARM member BBIISP00. The new value takes effect when the BBI-SS PAS restarts.

By default, the index table uses 200K of BBI-SS PAS extended private area storage. You can use the CBSORTSZ in BBPARM member BBIISP00 to change this value. Each terminal entry requires 12 bytes (8 bytes for the terminal ID and 4 bytes for the address of either the CNT or the CLB control block).

**Select Code:** OSTAT

**Parameter:** The following parameters can be used with this service. Multiple parameters must be separated by a comma (.).

COUNT>n

Where n specifies the number of messages queued. This keyword is used to display only those transactions that have more messages queued than the number specified for COUNT; for example:

COUNT>5.

Only transactions having more than five queued messages are displayed.

ID

Enter an 8-character user ID or a qualified user ID. OSTAT displays only those terminals that have a matching user ID, for example ID=U++D.

LTERM|LT

Enter an 8-character LTERM name or a qualified name. OSTAT displays only those terminals that have a matching name, for example LTERM=A1Q\*.

NONZERO|NZ

This keyword can be used to display only LTERMs with one or more messages queued for them. The format of this type of request is:

LTERM=L\*, NONZERO.

This example shows all LTERMs whose name begins with L and that have a message queued for output.

SEV1|SEV2

Shows only those terminals with the specified status; for example:

SEV1

Shows only terminals with SEV1 status.

START

Enter a starting LTERM to page through LTERMs starting with a particular LTERM, for example START=BOLD0.

**Scrolling:**

The START= field in the OSTAT display contains the name of the first terminal in the next list to be displayed. Use this field to scroll through the list either forward or backward. Entering A1QFFO1A, for example, puts that terminal name at the top of the next displayed list.

Pressing ENTER repetitively scrolls forward a page at a time. For example, if the START= field is BOLD1 as shown in the sample display above, pressing ENTER puts BOLD1 at the top of the next list of displayed terminal names.

To scroll to another place on the list, enter the name of a terminal in the START= field. A \* qualifier can be used to indicate a generic terminal name. For example, enter A\* to scroll to the beginning of the list.

**Field Descriptions:**

Each of the fields is shown and described below by display area.

Area 1: Line 5 of the OSTAT Display

IMS-VTAM INTERFACE NORMAL

IMS STATUS NORMAL

This area shows the status of the IMS-VTAM interface and IMS DC status. The descriptions are arranged in alphabetical order.

The IMS-VTAM interface status is shown in the left column. Possible statuses include:

- **CLOSE VTAM ACB PENDING**

IMS VTAM ACB is being closed or is pending. IMS VTAM communication is impossible.

- **IMS-VTAM INTERFACE IS NORMAL**

The IMS-VTAM interface status is normal.

- **SYSTEM NOT ACCEPTING LOGONS**

IMS is not accepting LOGON or /OPNDST.

- **TPEND HALT CANCEL PENDING**

Z NET,CANCEL for VTAM shutdown is pending and IMS VTAM communication is impossible.

- **TPEND HALT NON-QUICK PENDING**

Z NET for VTAM shutdown is pending and IMS VTAM communication is impossible.

- **TPEND HALT QUICK PENDING**

Z NET,QUICK for VTAM shutdown is pending and IMS VTAM communication is impossible.

- **VTAM ACB NOT OPEN**

IMS VTAM ACB is not open and IMS VTAM communication is impossible.

The IMS DC status is shown in the right column of this area. Possible statuses include:

- **/CHE DUMP QUEUES IN PROGRESS**

IMS is shutting down and /CHE DUMPQ is in progress.

- **/CHE DUMPQ QUIESCE ENTERED**

/CHE DUMPQ QUIESCE was entered. IMS is preparing for shutdown.

- **/CHE FREEZE QUIESCE ENTERED**

/CHE FREEZE QUIESCE was entered. IMS is preparing for shutdown.

- **/CHE PURGE IN PROGRESS**

IMS is shutting down and /CHE PURGE is in progress.

- **/CHE PURGE QUIESCE ENTERED**

/CHE PURGE QUIESCE was entered. IMS is preparing for shutdown.

- **CIOP POOL SHORT ON STORAGE**

IMS is in selective dispatching because of CIOP pool shortage.

- **CONTROL REGION IS ABENDING**

IMS is abending.

- **DISCONNECT ALL LINES**

IMS is shutting down and disconnecting all lines.

- **IMS SHUTDOWN IN PROGRESS**

IMS is shutting down and /CHE FREEZE is in progress.

- **IMS STATUS NORMAL**

IMS is functioning normally.

- **SEL DSP - NO DYNAMIC SAPS**

IMS is in selective dispatching because of dynamic SAP shortage.

- **/STA DC NOT PERFORMED**

/STA DC was not performed. IMS logon is not possible.

- **SYSTEM SHUT - QUIESCE MSG RGN**

IMS is shutting down and quiescing all MPP regions.

- **SYSTEM SHUT - STOP OUTPUT**

IMS is shutting down and all message output is stopped.

- **SYSTEM SHUT - TERM MSG RGNS**

IMS is shutting down and terminating all MPP regions.

- **SYSTEM SHUT - WAITING FOR DC**

IMS is shutting down and waiting for IMS DC to complete all message output in progress.

**Area 2:** Line 6 to the end of the OSTAT display

- LTERM -	- SIGNON -	NODE/SP	TERM	OUTPUT	LAST	
-- NAME --	--- ID ---	LN#/PTRM	TYPE	MSG #	TRN/DEST	TERMI NAL STATUS-----
A1QAAI 1A		L1WA1I 01	SUBPL	0		TERMI NAL I S NOT ALLOCATED
A1QAA01A		L1WA1001	SUBPL	0		TERMI NAL I S NOT ALLOCATED
A1QBBI 1A		L1WA1I 02	SUBPL	0		TERMI NAL I S NOT ALLOCATED

This area shows the IMS LTERMs and their output status. The descriptions are arranged in alphabetical order.

#### LAST TRN/DEST

The last input transaction code or preset destination name generated with the /SET command.

#### LTERM NAME

The name of the logical terminal (LTERM/CNT).

#### NODE/SP LN#/PTERM

The node name if the associated terminal is a VTAM device; the subpool name if the LTERM is part of a subpool and is not allocated yet; the line number and PTERM number if the associated terminal is a BTAM device.

#### OUTPUT MSG #

The number of output messages currently queued to this LTERM.

#### SIGNON ID

If a /SIGN ON command was entered at the terminal, this is the ID of the user.

**Note:** The signon ID is the security user ID, not the ETO user structure name (SPQB).

#### TERM TYPE

Device type of the associated terminal as defined in the Stage 1 system generation.

For unallocated ETO devices, ETO is shown.

#### TERMINAL STATUS

See “Output LTERM Status” on page 412.

---

## Output LTERM Status

Workload Analyzer scans all LTERMs for output status and reports the LTERMs that match the user-specified search criteria. OSTAT sorts the LTERM status by severity and reports the most significant ones. If an LTERM passes all checks, the status message is:

TERMINAL STATUS NORMAL

The following LTERM status messages are listed alphabetically. The severity is shown in parentheses to the left of each message.

SEV1 is the most serious.

SEV2 is less critical.

SEV3 is usually informational only.

- **(SEV2) ASYNC ERR MSG IN PROCESS**

For an LU6 session, the error message FMH is being sent.

- **(SEV2) CANCEL RECEIVED/SCHEDULED**

IMS received a request from the other LU6 half-session partner to cancel the input in progress or vice versa.

- **(SEV2) CLB DECB INVALID**

ECB for line is invalid. This is temporary while IMS is relocating the input message to the message queue.

- **(SEV2) CLSDST REQUESTED**

This terminal is requesting to terminate its session with IMS. This node will be disconnected.

- **(SEV2) CONVERSATION HELD**

The terminal had a /HOLD command issued to suspend and save a conversation. Conversation will be resumed (/RELEASE command) or terminated (/EXIT command) sometime later.

- **(SEV2) DEAD LETTER**

Messages queued to this terminal exceeded the ETO dead letter timeout limit.

- **(SEV2) DEQUEUE COMMAND FOR MSG**

The message IMS was preparing to send to the terminal was the target of a /DEQ command from the MTO.

- **(SEV2) DEQUEUE PURGE IN PROGRESS**

The terminal had a /DEQUEUE LINE or NODE PURGE command issued against it. All output is cancelled.

- **(SEV2) /DEQUEUE PURGE ISSUED LTERM**

A /DEQUEUE LTERM PURGE command was issued for the LTERM. All output is cancelled.

- **(SEV1) DFSCVCLO REQUEST IDLE LINE**

The IMS LOGON or OPNDST processor (DFSCVCLO module) requests that the line be idled.

- **(SEV1) DO NOT SEND QUEUED MSG**

The line or node associated with this LTERM was /PSTOPped.

- **(SEV2) DO NOT SEND QUEUED OUTPUT**

The physical terminal associated with this LTERM is currently in receive mode. IMS will not send queued output to this terminal until input is received. The input can be in the form of a PA1, PA2, another input message, or a PF key.

- **(SEV2) DYNAMIC TERMINAL NOT ALLOCATED**

This terminal was created dynamically by ETO and is not allocated to a node. This can occur under several circumstances; for example:

- Output was sent to this LTERM and ETO created the LTERM and user structure but not the node. If a node cannot be or is not created and allocated to this LTERM, messages are categorized as dead letters.
- A user logged off the system, either intentionally or through a network failure, and the status was such that the user structure and LTERM could not be deleted. This can occur, for example, if there are messages still queued to the LTERM or the user has an active conversation.

- **(SEV1) ERROR ON GET NEXT FOR OUTPUT**

IMS was preparing an output message for transmission. The first segment of the message was successfully retrieved from the queue, but one of the subsequent segments could not be accessed. This is a temporary error condition. IMS terminated the processing of this message, reset this status message, and marked the LTERM as having a queue error. See the I/O ERROR OCCURRED ON QUEUE message for the subsequent status.

- **(SEV2) FATAL ERP**

IMS has received or is scheduling an UNBIND for this LU6 ISC session.

- **(SEV1) /IDLE WITHOUT SHUTDOWN**

The terminal had an /IDLE LINE x PTERM y NOSHUT command issued against it. All input and output is terminated. The terminal remains IDLE until it is restarted.

- **(SEV1) IMS SHUTDOWN MSG BEING SENT**

IMS is shutting down. A shutdown message was sent to this terminal.

- **(SEV1) I/O ERROR OCCURRED ON QUEUE**

Some of the messages queued to this terminal are unavailable for transmission because I/O errors occurred on one or more of the queue data sets.

- **(SEV2) INPUT ONLY DEVICE**

This terminal is INPUT only. No output is possible for this device.

- **(SEV3) INQUIRY ONLY CNT**

This LTERM is defined as INQUIRY only.

- **(SEV1) /LOCK ISSUED FOR TERMINAL**  
The terminal is LOCKed. The remote terminal user locked the terminal with the /LOCK NODE or PTERM command. IMS does not receive nor send messages to a terminal that is LOCKed; however, it continues queuing output messages to the terminal.
- **(SEV1) /LOOPTEST IN PROGRESS**  
The terminal had a /LOOPTEST LINE x PTERM y command issued against it. The terminal is in an output write loop until an /END command terminates the loop test mode.
- **(SEV1) LOOPTEST IS PENDING**  
The terminal had a /LOOPTEST LINE x PTERM y command issued against it. The terminal will be in an output write loop until an /END command terminates the loop test mode.
- **(SEV2) LOST TERM**  
The IMS lost terminal exit was entered for this terminal because of the loss of communications with the terminal.
- **(SEV2) LTERM HAS PAGED MSG IN PROCESS**  
This LTERM is looking at a multi-page output message. The user can get to the end of the message or next message by using the PA1 and PA2 keys.
- **(SEV1) LTERM LOCKED FROM /LOCK**  
The LTERM is LOCKed. The remote terminal user locked the terminal with the /LOCK LTERM command or the MTO issued the /LOCK LTERM x. IMS does not receive nor send messages to an LTERM that is LOCKed; however, it continues queuing output messages to the terminal.
- **(SEV1) LTERM STOPPED FOR QUEUING**  
The LTERM is STOPped. IMS will no longer queue output messages to this terminal. Messages already queued will not be sent.
- **(SEV1) LTERM STOPPED FOR SENDING**  
The LTERM is PSTOPped. IMS will continue queuing messages to this LTERM, but none will be sent.
- **(SEV2) MSC LCB NOT ALLOCATED**  
LTERM is for MSC (Multiple System Coupling), but the LCB (Link Control Block) has not been allocated to the LTERM yet. Use /START MSNAME to start the link.
- **(SEV3) MSC LINK CNT REMOTE=xxx**  
This LTERM is for MSC (Multiple System Coupling). xxx is its partner SID.
- **(SEV2) NOT READY FOR OUTPUT**  
The associated VTAM node is not ready to accept output. There is at least one message queued to this LTERM, but other asynchronous I/O is already active against the node. IMS will not send the queued output until the node completes its current communication.
- **(SEV1) OUTPUT QUEUING STOPPED**  
The node or line associated with this LTERM was the target of a /STO command. IMS will not queue messages to this LTERM until a /STA NODE or a /STA LINE command is issued.

- **(SEV1) OUTPUT STOPPED ON THIS LINE**  
This line has been stopped for output, but queuing continues.
- **(SEV1) P-LINK NO MORE LOGON**  
This LTERM is for the MSC (Multiple System Coupling) whose physical link is not accepting any more LOGONs.
- **(SEV1) P-LINK NOT OPEN**  
This LTERM is for the MSC (Multiple System Coupling) whose physical link is not yet opened.
- **(SEV1) QUEUE ERROR WITH SYSMMSG**  
For LU6 sessions, an I/O error on the IMS queues is the cause of the SYSMMSG FMH that was sent to the partner half-session.
- **(SEV1) QUEUING IS STOPPED**  
The terminal is stopped for all message queuing. No output messages will be queued to the terminal.
- **(SEV1) /QUIESCE ISSUED**  
The node associated with this LU6 session was /QUIesced by the MTO.
- **(SEV2) RECOVERY REQUEST ON THIS NODE**  
For IMS SLU type P sessions (LU 0), IMS is attempting to resynchronize the session after protocol errors occurred.
- **(SEV3) REJECT IF NOT INQUIRY TRAN**  
IMS rejects the transaction from this terminal because the transaction was not defined as inquiry only.
- **(SEV3) REMOTE CNT**  
This terminal is a remote-CNT for MSC link.
- **(SEV2) RESTART RECEIVED/SCHEDULED**  
IMS is in the process of attempting to re-establish an LU6 session or the other half-session is attempting to restart the session with IMS.
- **(SEV2) SCHED/STACK LOGON**  
IMS was asked to issue an open destination to this terminal.
- **(SEV2) SCHEDULE CANCEL**  
For LU6 sessions, a CANCEL FMH must be sent.
- **(SEV3) SHARED LTERM**  
This LTERM is defined as a shared terminal.
- **(SEV1) SNA NODE SENT QUIESCE**  
This terminal is requesting to terminate its session with IMS. This node will be disconnected.
- **(SEV3) SUBPOOL CNT**  
This LTERM is part of a subpool.

- **(SEV1) SUBPOOL STOPPED**  
CNT subpool is stopped.
- **(SEV1) SYSTEM SHUTDOWN PENDING**  
IMS is shutting down; no more input is allowed from the terminal.
- **(SEV1) TERMINAL DD EXTENT MISSING**  
An attempt to open a line group failed because the DD extent is missing.
- **(SEV2) TERMINAL IS AWAITING RESPONSE**  
This terminal sent input to IMS and is waiting for acknowledgement. The terminal is defined as FORCRESP or TRANRESP, or a Fast Path transaction was entered.
- **(SEV3) TERMINAL IS IN A CONVERSATION**  
The terminal entered a conversational transaction and is currently in the middle of that conversation.
- **(SEV1) TERMINAL IS INOPERABLE**  
The associated physical terminal is inoperable. IMS detected some problem with the terminal and marked it inoperable; however, IMS continues queuing output messages to the terminal.
- **(SEV2) TERMINAL IS NOT ALLOCATED**  
The terminal is not allocated. This is a VTAM subpool LTERM that is subject to dynamic allocation and is not currently in session. The node and the user ID are both set to NONE.
- **(SEV3) TERMINAL IS NOT CONNECTED**  
The terminal is not connected to IMS. The user can log on a VTAM terminal using the site's LOGON APPLID or the MTO can issue an /OPN NODE xxxx. If the terminal is a dial-up, the user can call in to IMS.
- **(SEV1) TERMINAL IS STOPPED FOR OUTPUT**  
The terminal had a /STOP LINE or NODE command issued against it. No output will be sent by IMS until a /START command is issued.
- **(SEV2) TERMINAL IS WAITING FOR RESP**  
This terminal sent input to IMS for a response-mode transaction. The terminal is not available, because IMS did not respond.
- **(SEV2) TERMINAL NOT SIGNED ON**  
This terminal requires that the operator sign on before using it. The operator should issue a /SIGN ON command, a user ID, and an optional site-dependent password.
- **(SEV1) /TEST ISSUED FOR TERMINAL**  
The terminal was placed in test with the /TEST command. It is in test until /END is issued.
- **(SEV2) THIS LINE IS INACTIVE**  
The line or node associated with this LTERM is inactive, because a session was never started or there are no messages queued.
- **(SEV3) VTAMPOOL CNT**  
This LTERM is defined as part of the VTAM pool to support the LU6 (ISC) session.

- **(SEV2) WAIT FOR LUS-ABORT**

The LU6 session is in the process of being ABORTed. IMS is waiting for the LU ABORT status FMH.

- **(SEV1) WAITING FOR LOSTERM**

IMS lost communications with this VTAM terminal and is waiting for VTAM to invoke the lost terminal exit.

- **(SEV2) WAITING FOR L1-LATCH (xxxx)**

The terminal output ITASK is waiting for Level 1 Latch. xxxx is the name of the latch.

- **(SEV2) WAITING FOR L2-LATCH (xxxx)**

The terminal output ITASK is waiting for Level 2 Latch. xxxx is the name of the latch.

- **(SEV2) WAITING FOR POOL SPACE (xxxx)**

The terminal output ITASK is waiting for pool space. xxxx is the name of the pool.



## Chapter 20. TRANQ - Transaction Queue Status Display

```

BMC SOFTWARE ----- TRANSACTION QUEUE STATUS ----- PERFORMANCE MGMT
SERV ==> TRANQ   INPUT   14: 40: 23   INTVL=> 3   LOG=> N   TGT==> IMSA
PARM ==> THDAMI NQ

TRANCODE  S   COUNT  CLS PR  PSBNAME  QLI M  PRCL  PARL  TRANSACTION STATUS
-----
DSPALLI   1     0    1   7  DFSSAM07    3  65K    0 PSB IS TP PGM
DSPINV    1     0    1   7  DFSSAM03    3  65K    0 PSB IS TP PGM
FPDUMMY   1     0    1   1  DBFDUMMY   65K  65K    0 PSB DIR INIT FAILED
FPSAMP1    1     0    1   1  DBFSAMP3   28K  65K    0 SMB FASTPATH EXCLUSIVE
FPSAMP2    1     0    1   1  DBFSAMP4   65K  65K    0 PSB IS TP PGM
PART      1     0    1   7  DFSSAM02    3  65K    0 PSB IS TP PGM
STARTR    3     0    1   5  SAG600     10  10     0 PSB DIR INIT FAILED
TBMP1     1     0    4   0  PBMP1      65K  65K    0 PSB IS BMP PGM
TBMP1V    1     0    4   0  PBMP1V     65K  65K    0 PSB IS BMP PGM
TBMP2     1     0    5   0  PBMP2      65K  65K    0 PSB IS BMP PGM
TCON00    3     0    3   5  PCON00     10  10     0 SMB CONVERSATIONAL
TCON01    3     0    3   5  PCON01     10  10     0 SMB CONVERSATIONAL
TDRI VER  1     0    4   0  PDRI VER   65K  65K    0 PSB IS BMP PGM
TE4CNI NQ 1     0    1   1  PE4CNI NQ  65K    5     0 PSB IS TP PGM
TE4COCNG  1     0    1   1  PE4CODEL   65K    5     0 SMB CONVERSATIONAL
TE4CODEL  1     0    1   1  PE4CODEL   65K    5     0 SMB CONVERSATIONAL
TE4COINQ  1     0    1   1  PE4COINQ   65K    5     0 PSB IS TP PGM
TE4CONEW  1     0    1   1  PE4CORDR   65K    5     0 SMB CONVERSATIONAL

```

### Description:

The transaction queue status display shows the status of transactions in the IMS input queue or scheduler message block (SMB) queues. It also shows the number of transactions in the input queue for any IMS transaction code.

### Select Code:

TRANQ

### Parameter:

The following parameters can be used with this service. Multiple parameters must be separated by a comma (.).

COUNT>n

Where n specifies the number of messages queued. This keyword is used to display only those transactions that have more messages queued than the number specified for COUNT; for example:

COUNT>5

Only transactions having more than five queued messages are displayed.

NONZERO|NZ

This keyword is used to display only transaction codes with one or more messages queued for them. For example:

TRAN=A\*, NONZERO

Displays all transactions beginning with an A character that have messages queued for processing.

#### SEV1|SEV2

Shows only those transactions with the specified status; for example:

#### SEV1

Shows only those transactions with SEV1 status.

#### SSQ

Shows only those transactions scheduled in a region that is waiting on a subsequence queue.

#### START

Enter an 8-character transaction code or a qualified transaction code to begin paging through the transaction codes starting with that particular name, for example  
START=TE4CONEW

**Note:** For dynamic transaction codes, the exact transaction code must be specified.

#### SUSP

Shows only those transactions that have messages on the suspend queue.

#### **Tuning Tip**

Suspended messages are a problem with response mode transactions, because users requesting the transaction code cannot issue any more transactions until their message is removed from the suspend queue.

Messages are typically put on the suspend queue when a transaction attempts to access an unavailable resource such as a database. IMS pseudo-abends the transaction with a U3303 and places the input message on the suspend queue. If the suspend queue becomes too long, IMS puts the transaction into a USTOP status to prevent more transactions from being scheduled. The suspended condition can be cleared by making the resource identified in the DFS3303 message available. For example, if a database is not available because it is stopped, you can clear the suspend condition by starting the database.

#### TRAN

Enter an 8-character transaction code or a qualified transaction code. TRANQ displays only those transactions that have a matching TRANCODE, for example  
TRAN=TE4\*.

TYPE=STAT|DYN

DYN

Shows only the transaction codes created dynamically for explicit APPC transactions.

STAT

Shows only the static transaction codes from the IMS Stage 1 definition (default).

**Scrolling:**

The START= field in the TRANQ display contains the first transaction code in the next list of transactions to be displayed. Use this field to scroll through the list either forward or backward. For example, entering BBFTRN19 puts that transaction at the top of the next displayed list.

Pressing ENTER repetitively scrolls forward a page at a time. For example, if the START= field is THDAMINQ as shown in the sample display above, pressing ENTER puts that transaction at the top of the next list of displayed transactions.

To scroll to another place on the list enter the name of a transaction in the START= field. The \* qualifier can be used to indicate a generic transaction code. For example, enter A\* to scroll to the beginning of the list.

**Field Descriptions:**

Each of the display fields is described below. The descriptions are arranged in alphabetical order.

CLS

Current class of the SMB (transaction). A remote transaction has a class of 0.

COUNT

Number of messages currently queued to this SMB (transaction).

PARL

The parallel limit is the number of messages that cause a transaction to be scheduled in another MPP region if the SMB (transaction) is defined as PARALLEL CAPABLE. It is defined in the IMS Stage 1 generation by the PARLIM operand of the TRANSACT macro contains SCHDTYP=SERIAL.

PR

Current scheduling priority of the SMB (transaction).

PRCL

The process limit is the number of messages that can be processed by a program in a single scheduling. It is defined in the IMS Stage 1 generation by the PROCLIM operand of the TRANSACT macro.

PSBNAME

Name of the program specification block (PSB) whose associated program processes this SMB (transaction).

#### QLIM

Queue limit count is the count of currently enqueued messages that cause the transaction's scheduling priority to change from normal to limit. The limit is defined in the IMS Stage 1 generation by the PRTY operand of the TRANSACT macro.

#### S

Transaction scheduling option (same as IMS Stage 1 generation, TRANSACT macro, SCHD parameter). If scheduling of this transaction failed because of an IMS internal reason, then one of the following applies:

- 1      Schedule transactions of equal or higher priority within the same class. This is the default.
- 2      Schedule transactions of higher priority within the same class.
- 3      Schedule any transaction within the same class.
- 4      Schedule next class of transaction.

#### TRANCODE

The name of the transaction (SMB) being displayed.

---

## Transaction Status

IMS transactions can be in one of several states. The TRANQ service tests the status of a transaction code in a predefined sequence so that the most critical information is always displayed.

The following status messages are listed in alphabetical order. The status category is listed in parentheses to the left of each message. The status categories are:

- SEV1 SMB (transaction) is unusable. This status is highlighted on the display.
- SSQ Transaction is queued and pending scheduling.
- SEV2 SMB (transaction) has a problem, but is still usable.
- SEV3 SMB (transaction) is normal.

- **(SSQ) BMP WAITING FOR BLOCK LOADER**

SMB (transaction) is scheduled for a BMP or JBP region, but the BMP or JBP is waiting in block loader for PSB (program) and/or DMB loading. The block loader can be active for only one region at a time.

If this status appears consistently, revise your IMS Stage 1 definition (RESIDENT and DOPT options of the APPLCTN macro).

- **(SEV1) DEST(RSMB) Q ERROR (SYSID=XXX)**

The transaction queued from the SMB is to be sent to another IMS (SYSID=XXX). However, the target SMB has a queue error, so the target IMS cannot process the transaction(s).

- **(SEV1) DO NOT SCHEDULE THIS PSB**

The PSB (program) defined to process this transaction is stopped and cannot process the transaction(s). Terminal input and queuing are still possible.

This can be caused by either a /STOP PGM command or a program abend. Use the /START PGM command to re-enable the PSB.

- **(SEV1) DO NOT SCHEDULE THIS SMB**

The SMB (transaction) has been stopped with the /PSTOP TRAN command. Terminal input and queuing are still possible. Use the /START TRAN command to re-enable the SMB (transaction).

- **(SSQ) MPR WAITING FOR BLOCK LOADER**

SMB (transaction) is scheduled for an MPP region, but the MPP is waiting in the block loader for PSB (program) and/or DMB loading. The block loader can be active for only one region at a time.

If this status appears consistently, revise your IMS Stage 1 definition (RESIDENT and DOPT options of the APPLCTN macro).

- **(SEV2) NO MPP SERVING CLASS**

No message region is assigned to process the class to which this transaction belongs. Either assign the transaction to a class that is being served by a region or assign the class to a new or existing region.

- **(SEV2) NO MPP SERVING CLASS/AGN**

ISI security is active, but there is no region to serve the application group name to which the transaction code belongs. You should start a message region that is assigned an AGN that includes this transaction code.

- **(SEV1) PSB DIR INIT FAILED**

The PSB (program) defined to process this transaction could not be initialized at IMS startup and therefore cannot process the transaction(s). Usually this is because the PSB (program) is not in the ACBLIB. Make an online change to switch to a new ACBLIB containing the PSB (program).

- **(SEV3) PSB IS BMP PGM (SCHEDULED)**

The PSB (program) defined to process this transaction is a BMP or JBP program. If SCHEDULED appears, the PSB (program) is loaded and can be scheduled.

- **(SEV1) PSB IS LOCKED**

The PSB (program) defined to process this transaction was locked by the /LOCK PGM command. Use /UNLOCK PGM to enable the PSB (program) to process transactions.

- **(SEV3) PSB IS TP PGM (SCHEDULED)**

The PSB (program) defined to process this transaction is an MPP or JMP program. If SCHEDULED appears, the PSB (program) is loaded and can be scheduled.

- **(SEV3) PSB IS TYPE 3 BATCH (SCHEDULED)**

The PSB (program) defined to process this transaction is a type 3 batch program. If SCHEDULED appears, it means the PSB (program) is loaded and can be scheduled.

- **(SEV3) REMOTE TRANSACTION (SYSID=XXX)**

The transaction is sent to another IMS (SYSID=xxx) to be processed.

- **(SEV3) SMB CONVERSATIONAL**

The transaction is conversational.

- **(SEV3) SMB ENQUEUED ON PST**

The SMB (transaction) is queued on a dependent region (PST) to be processed.

- **(SEV3) SMB ENQUEUED ON TCT**

The SMB (transaction) is queued on the transaction class table (TCT) waiting to be scheduled.

- **(SEV3) SMB FASTPATH EXCLUSIVE**

The transaction is Fast Path exclusive.

- **(SEV3) SMB FASTPATH POTENTIAL**

The transaction is Fast Path potential.

- **(SEV2) SMB IN/WAITING FOR BLK MOVER**

The SMB (transaction) is scheduled and waiting in the block mover for PSB relocation. The block mover can be active in only one region at a time.

- **(SEV1) SMB LOCKED**

The SMB (transaction) is locked by the /LOCK TRAN command and cannot be scheduled. Use /UNLOCK TRAN to enable the SMB (transaction) for scheduling.

- **(SEV1) SMB NOT SCHEDULED (MPP,PRI=0)**

The SMB (transaction) is defined to be an MPP transaction. It is not schedulable, because the priority is 0.

- **(SEV2) SMB PAR LIM REACHED (ZZ9 RGNS)**

The SMB (transaction) is currently scheduled in ZZ9 regions, but the number of transactions queued exceeds the parallel limit. Therefore it is eligible to be scheduled in another available region.

- **(SEV2) SMB PRIORITY RAISED FROM Z9**

The number of transactions enqueued on the SMB exceeded the priority limit (QLIM). IMS raised the priority of the SMB from Z9 (normal) to its current priority limit.

- **(SEV2) SMB SCHEDULE IN MAX ZZ9 RGNS**

The SMB (transaction) is defined to be scheduled concurrently in a maximum of ZZ9 regions and has now reached this limit.

- **(SEV2) SMB SCHEDULED IN ZZ9 REGIONS**

The SMB (transaction) is currently scheduled in ZZ9 regions.

- **(SEV1) SMB STOPPED FOR QUEUING**

The SMB (transaction) is stopped. Use /STA TRAN command to start it.

- **(SEV1) SMB STOPPED MSG SENT**

The SMB (transaction) is stopped and message DFS554I was sent to the IMS MTO, because its associated PSB (program) abended. The PSB (program) is also stopped. Use /START TRAN and /START PGM to restart the transaction.

- **(SEV1) SMB SUSPENDED WAITING**

The SMB (transaction) is suspended. This can be caused by a resource problem or IRLM or XRF takeover. A probable resource problem is that the database is not available. It is either stopped or locked, or could not be allocated.

After resolving the cause of the problem, use /START DATABASE to redrive the transactions.

**Note:** The MSG-Q-COUNT can be zero even though there is at least one transaction in the SUSPEND queue, because the SUSPEND queue is not part of the normal message queue.

- **(SEV1) SMB USTOPPED - DB NOT AVAIL**

The SMB (transaction) is U-STOPped and cannot be scheduled, because a database required for the PSB is not available. After resolving the problem, use /START TRAN to re-enable the SMB (transaction) for scheduling.

For some IMS versions, this status may not show. Instead, you may get SMB SUSPENDED WAITING (see this message definition in this list).

- **(SEV2) SMB WAITING CUTOFF PRIORITY**

The cutoff priority of the current transaction is lower than the cutoff priority for the current class.

- **UNKNOWN STATUS**

The status of the SMB (transaction) is unknown.

- **(SSQ) WAITING FOR DMB POOL SPACE**

DMB pool space is insufficient for IMS to complete SMB (transaction) scheduling. The SMB (transaction) is in a wait queue for retry later. If this status appears consistently, revise your Stage 1 definition, the DFSPRRG0 system options module, and/or the JCL to increase the DMB pool size.

- **(SSQ) WAITING FOR INTENT**

There is a database intent conflict between this transaction and another currently scheduled transaction. The SMB (transaction) is in a queue waiting for retry later.

- **(SSQ) WAITING FOR PSB POOL SPACE**

PSB pool space is insufficient for IMS to complete SMB (transaction) scheduling. The SMB (transaction) is in a queue waiting for retry later. If this status appears consistently, revise your Stage 1 definition, the DFSPRRG0 system options module, and/or the JCL to increase the PSB pool size.

- **(SSQ) WAITING FOR PSB WORK POOL**

PSB work pool space is insufficient for IMS to complete SMB (transaction) scheduling. The SMB (transaction) is in a queue waiting for retry later. If this status appears consistently, revise your Stage 1 definition, the DFSPRRG0 system options module, and/or the JCL to increase the PSB work pool size.

## Chapter 21. USER - User Status Summary

BMC SOFTWARE -----			IMS USER DISPLAY			----- PERFORMANCE MGMT		
SERV ==>	USER	INPUT	15:03:43	INTVL=>	3	LOG=>	N	TGT==> IMSA
PARM ==>		, SORT=UI			ROW 1 OF 3 SCROLL=> CSR			
EXPAND: OSTAT, ISTAT, TRANQ, APPCA								
USERID	USER	LTERM	NODE	ENQ	QCT	TRAN	AGE	STATUS
-----								
PWW1	PWW1	PWW1	LF05	0	0	PART	0	DYN
PWW3	N/A	USER1	LF04	0	0	N/A	0	STAT
		USER2		0	0			
***** END OF DATA *****								

**Description:** This display shows summary information for IMS users, both static and dynamic (ETO).

USER displays:

- Status of each user structure and its associated resources
- Security user ID (if available), which can be different from the user structure name for dynamic terminals
- The static users (users signed on to static terminals)

**Color:** If you have a color monitor, users in exception status (see the XON parameter description) are shown in red.

**Select Code:** USER

**Parameter:** All of the USER parameters, except for SORT, act as filters that restrict the information shown according to the criteria specified by the parameter(s). They can be used as follows:

- Multiple parameters must be separated by commas.
- A blank indicates the end of a parameter(s).
- Multiple resources with similar names can be requested by using an \* character as a generic qualifier and a + character as a positional qualifier. The positional qualifier is repeated for every character to be replaced. The generic qualifier replaces a group of characters. For example, a parameter of USERID=AB+D\* shows all USERIDs that start with AB, have D in the fourth position, and have any character following D.
- If multiple filtering parameters are entered, the users displayed must meet **all** of the restrictions.
- If one parameter invalidates another, an error message is issued without further processing.

The following parameter descriptions are arranged alphabetically. Parameters containing a numeric character are first in their alphabetical group.

#### ALLOC

Displays only those users allocated to a node. NALLOC cannot be used when ALLOC is specified.

#### CONV

Displays users with an active conversation.

#### DEADQ

Displays users with queues marked as dead letter queues.

#### DYN

Displays only dynamic users. STATIC cannot be used when DYN is specified.

#### EXCL

Displays users in exclusive mode.

#### HELD

Displays users with a held conversation.

#### ISC=NO | YES

Displays static ISC users only when ISC=YES is specified.

#### L=value

Locates the specified value where value applies to the current SORTed column. That row is placed at the top of the list. Generic and positional qualifiers cannot be used for this parameter.

#### LTERM=l term

Where lterm is a 1- to 8-character LTERM name. Generic and positional qualifiers can be used for the LTERM name. Displays users with a matching LTERM name. To display rows that do not have an LTERM name, specify LTERM=' ' or LTERM=N/A.

#### MFST

Displays users in MFS test mode.

## NALLOC

Displays users not allocated to a node. ALLOC cannot be used when NALLOC is specified.

## NODE=i d

Where i d is a 1- to 8-character node ID. Generic and positional qualifiers can be used for the node ID. Displays users with a matching node ID. To display rows that do not have a node ID, specify NODE=' ' or NODE=N/A.

## NZ

Displays users with a nonzero queue count.

## PRST

Displays users in preset mode (IMS /SET command issued).

## QCT> nnn

Where nnn can be 1 to 999. Displays users with a queue count exceeding nnn.

## RESP

Displays users in response mode.

## RESPINP

Displays users in response-input mode.

## SORT|SO=cc

Where cc can be any of the following two characters. The display list is sorted by USERID by default.

AG       Sorts the list by the number of days that the oldest message was enqueued to the LTERM (descending).

EN       Sorts the list by ENQ count (descending).

LT       Sorts the list by LTERM (ascending).

NO       Sorts the list by NODE (ascending).

QC       Sorts the list by the queue count (descending).

TR       Sorts the list by trancode name (ascending).

UI       Sorts the list by USERID (ascending is the default).

US       Sorts the list by USER (ascending).

## STATIC

Displays only static users. DYN cannot be used when STATIC is specified.

## STOP

Displays stopped users.

## TRAN=trancode

Where `trancode` is a 1- to 8-character transaction name. Generic and positional qualifiers can be used for the transaction name. Displays users whose current or last transaction have a matching transaction name. To display rows that do not have a transaction name, specify `TRAN=' '` or `TRAN=N/A`.

## TST

Displays users in test mode.

## USER=user

Where `user` is a 1- to 8-character user name. Generic and positional qualifiers can be used for the user name. Displays only those users that have a matching name. To display rows that do not have a user name, specify `USER=' '` or `USER=N/A`.

## USERID=i d

Where `i d` is a 1- to 8-character user ID. Generic and positional qualifiers can be used for the user ID. Displays only those user IDs that have a matching ID. To display rows that do not have a user ID, specify `USERID=' '` or `USERID=N/A`.

## XON

Displays users with **any** of the following:

CONV  
DEADQ  
EXCL  
MFST  
STOP  
TST

### **Expand:**

USER can be EXPANDED by moving the cursor to the following fields and pressing ENTER:

OSTAT	Terminal output status display
ISTAT	Terminal input status display
TRANQ	Transaction queue status display
APPCA	APPC activity display

**Sorting:**

The display list can be sorted by:

- Using the SORT parameter.
- Moving the cursor to the column heading to be sorted and pressing ENTER.

This overrides any SORT parameter entered in the PARM field and primes the field with the action taken. The integrity of any other parameters entered in the PARM field is preserved.

The display list is sorted by USERID by default. Alphanumeric fields are sorted in ascending order. Numeric fields are sorted in descending order. Fields with no values are sorted to the bottom of the list.

**Scrolling:**

The display is scrollable.

**Field Descriptions:**

Each of the fields is described below. The descriptions are arranged in alphabetical order.

**AGE**

Age in days of the oldest message enqueued to the LTERM. An \*\* in this column indicates the age of the oldest message is over 99 days. STATUS field for this user contains DEADQ.

**ENQ**

The number of messages enqueued to the LTERM displayed in this row.

**LTERM**

Logical terminal name assigned to this user or node.

If a user or node has multiple LTERMs assigned to it, the LTERMs appear on separate rows with their own ENQ and QCT values. USERID, USER, and NODE are blank if they are the same as the row above.

**NODE**

The signon or allocated node name for this user. This field contains N/A if the user is not assigned to any node (dynamic only).

**QCT**

The sum of messages still on the queue for this LTERM.

**STATUS**

The status of this user. For static terminals, this is the status of the node associated with the user. Multiple status codes are displayed within the space limitations of the display line. The status can be:

**CONV**

The user is in an active conversation.

**DEADQ**

The user has messages on the queue that have exceeded the dead letter timeout value.

**DYN**

The user was created by ETO signon.

EXCL	The IMS /EXCLUSIVE command was issued for this user. This user only receives messages from transactions submitted by this physical terminal. Other messages remain queued until the EXCL status is terminated with an IMS /END or /START command.
HELD	The user has a held conversation.
MFST	Transactions for this user search the MFS test library first for formats.
NALLOC	The user is not allocated to a VTAM node.
PRST	The user is in preset mode (IMS /SET command issued).
RESP	The user is in response mode.
RINP	The user is in response-input mode.
STAT	The user structure was prebuilt; for example, defined in Stage 1.
STOP	The user is stopped. With this status, the user cannot sign on to the IMS system (ETO only).
TST	The user is in test (echo) mode.
USER	Name of the user structure (SPQB) either created dynamically for a dynamic user or already existing (through system generation) for an ISC type connection.
USERID	Name of the security user ID.
TRAN	The name of the last or current transaction code processed at this node.

---

## Chapter 22. Requesting Workload Wait Data Collection (MWAIT)

The Workload Analyzer Workload Wait service identifies and displays the events that account for a transaction's response time in IMS. The display of workload wait events requires previous data collection.

IMS workload wait data is collected by MWAIT. MWAIT samples workloads waiting for resources. The MWAIT data collection service periodically samples IMS and accumulates statistics for the requested workloads. This statistical data is displayed online by using the DWAIT service. Data collection must first be activated through the MWAIT service before the data can be displayed by the DWAIT service.

**Note:** For MWAIT to collect DBCTL thread data, `CI CS=YES` or `CI CS=ONLINE` must be specified in BBPARM member IMFECP00.

---

### Starting MWAIT

MWAIT data collection can be activated by:

- Selecting the MWAIT service from the list of monitor data collection services (SM command)

Use the S line command from the Data Collection Monitors service application (see *Monitors and Traces Reference Manual*).

A data entry panel, shown on page 435, is displayed so that you can specify request options, described in Table 5 on page 436:

- Pressing ENTER activates a request.
- Pressing END redisplay the monitor service list.

Multiple wait analysis requests with different selection criteria can be run concurrently. Each MWAIT request can be given an optional name for identification by specifying a 1- to 8-character identifier in the PARM field. One MWAIT request can be activated without an identifier; the request identifier defaults to blanks.

For example, this MWAIT data collection request for the transaction PART is identified as MON1:

```
PARM      ==> MON1
INTERVAL  ==> 200MSEC
TRAN      ==> PART
```

If no other options are specified and all the defaults are taken, this request collects data for the total IMS workload (all transactions processed), using the default Timer facility sampling interval. When the collected data is displayed using the DWAIT service, the default current period (CURPER) and history definition are used.

This request:

```
INTERVAL ==> 200MSEC
CURPER   ==> 00: 10: 00
HI STORY ==> 02: 00: 00
```

collects data for the total IMS workload. Sampling will be performed every 200 milliseconds. When the collected data is displayed using the DWAIT service, the current period (CURPER) is 10 minutes and the history is 2 hours.

This request:

```
PARM     ==> SAMPLE
PSB       ==> SPT+++ BHR+++
REGION   ==> CICSPRD1
```

collects data for all transactions executed in region CICSPRD1 whose PSB begins with the letters SPT or BHR.

The selection keywords are ANDed; keyword operands are ORed.

- Replicating an active request from the Active Timer list application (Primary Menu Option 2) as described in the chapter “Displaying a List of Active Timer Requests” in *Monitors and Traces Reference Manual*.

You can replicate a request by using the R line command in the Active Timer list.

- Use the D line command from the Data Collection Monitors service list application to access the Active Timer list for only the selected service.
- Access the Active Timer list application directly from the Primary Option Menu (Primary Menu Option 2).

- Starting the MWAIT service from BBPARM with other service requests

Define a series of SET requests as a member of your BBI-SS PAS BBPARM data set that can be started automatically when the system starts or at your request (see the “Set Timer Requests” chapter in *Monitors and Traces Reference Manual*).

- Starting MWAIT from an MAINVIEW AutoOPERATOR EXEC

Write an EXEC that starts MWAIT (MAINVIEW AutoOPERATOR must be installed). Use the IMFEXEC IMF command followed by the service name, optional parameters, and an identifier for the target system; for example:

```
IMFEXEC IMF SET REQ=MWAIT TRAN=PAY+ TARGET=IMSVSx
```

For more information, see the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs*.

- Starting MWAIT with a SET request

Define a SET request for MWAIT from the SERV field of any display.

---

## Using the Workload Wait Data Entry Panel

The Workload Analyzer Wait (MWAIT) monitor is a monitoring service that collects the workload wait events for all or specific IMS workloads. To display the data entry panel, shown in Figure 3 on page 435, for requesting the MWAIT monitor service:

1. Enter an SM command as described in the chapter called “Displaying a List of Data Collection Monitors” in *MAINVIEW for IMS Online – Monitors and Traces Reference Manual*.
2. Use an S line command to select MWAIT from the services shown in the display list of monitor services.

BMC SOFTWARE	-----	START	WA WAIT REQUEST	-----	PERFORMANCE MGMT
COMMAND	====>				TGT ==> IMST
MWAIT - MONITOR WORKLOAD WAIT					
PARM	==>		(Identifier)		
INTERVAL	==> 200MSEC	START ==>	STOP ==>		QIS ==> YES
RST	==> HOT		(Restart Option: HOT, COLD, PUR, QIS)		
CURPER	==> 00: 01: 00		(Current Period (hh: mm: ss))		
HISTORY	==> 00: 30: 00		(History Period (hh: mm: ss))		
Specify Workload Selections:					
PGMTYPE	==>		(MPP, MDP, I FP, FPU, TPI, BMP, DBT, NOTDBT)		
TRANTYPE	==>		(ALL, DLI, DB2, FP)		
TRAN	==>				
CLASS	==>				
PROG	==>				
PSB	==>				
REGION	==>				
RGNID	==>				
TERM	==>		(name, MSCCLOCK, SYNCLOCK)		
USERID	==>				

Figure 3. Start Workload Wait Request

These options are SET keywords for MWAIT and are described in Table 5 on page 436 in alphabetical order.

Table 5. SET Keywords to Activate a Workload Wait (MWAIT) Service

Keyword	Operand	Description
CURPER CP	hh:mm:ss	<p>Specifies the interval for the CURRENT PERIOD area of the DWAIT display produced by Workload Analyzer. At the end of this interval, the sample counters are reset to zero.</p> <p><b>Note:</b> The CURRENT PERIOD interval must be an integer multiple of the INTERVAL value; if not, it is rounded up.</p> <p>Default is 30 seconds (00:00:30 - specified in BBPARM member BBIISP00).</p>
HISTORY H	hh:mm:ss	<p>Specifies the interval for the HISTORY area of the DWAIT display produced by Workload Analyzer. At the end of this interval, the sample counters are reset to zero.</p> <p><b>Note:</b> The HISTORY interval must be an integer multiple of the CURPER interval; if not, it is rounded up.</p> <p>Default is 12 hours (12:00:00 - specified in BBPARM member BBIISP00).</p>
INTERVAL I	hh:mm:ss	<p>The time interval between successive invocations of the requested service. The default is one minute (00:01:00) or as specified by the user in the BBIISP00 member of the BBPARM data set.</p> <p>It can be used with the LOG keyword to request automatic logging of a display to the BBI-SS PAS Image log.</p> <p><b>Note:</b> The MOD keyword cannot be used to change this option. The request must be purged and a new request must be made.</p>
PARM	i d	<p>Where i d is a unique 1- to 8-character identifier. If you want to run multiple requests, specify an ID in the PARM field to make this request unique. Blank can also be used as an identifier for one request. The identifier is used to specify which monitor data is to be displayed with the plot display service.</p>
QIS		Defines the action to be taken for the service when IMS is not active.
	YES	Specifies that the service is to be quiesced. This is the default.
	NO	<p>Specifies that the service is to start or continue running.</p> <p><b>Note:</b> When QIS=NO is specified, monitors that require IMS continue to be scheduled at each interval; however, they return zero values. Any analyzer services set up to run asynchronously fail with a short message of CANNOT LOCATE IMS SPECIFIED in the first line. The BBI-SS PAS Image log contains screen images of these services.</p>

Table 5. SET Keywords to Activate a Workload Wait (MWAIT) Service (continued)

Keyword	Operand	Description
RST		Defines the restart option to be used when a service is quiesced because of an inactive IMS subsystem or RRR is specified for BLK (see the “Set Timer Request” chapter in <i>Monitors and Traces Reference Manual</i> ). Default is HOT.
	HOT	Restarts the service automatically without loss of previous data.
	COLD	Restarts the service automatically; all previously collected data is deleted.
	PUR	Purges the service automatically when the target IMS starts.
	QIS	Keeps the service in a quiesced state until it is purged by an authorized user.
START	hh:mm:ss	<p>Requests monitor start time. If the time entered is more than 10 minutes prior to the current time, 24 hours are added to the specified time and the request is started the next day. To start a request at midnight, specify 24:00:00.</p> <p>The default is the next full minute.</p> <p><b>Note:</b> This option cannot be modified. The request must be purged and a new request must be made.</p>
STOP	nn   hh: mm: ss	<p>Requests monitor stop limit where nn is length of time in minutes and hh: mm: ss: is a timestamp.</p> <p>Processing stops at the end of the last interval before the specified stop time. This time is displayed in the STOP field when the request is viewed with the R, P, M, or W line commands from the Active Timer Requests list (Primary Menu Option 2) application.</p> <p>If the time entered is the same as the START time, 24 hours are added to the STOP time.</p>
<b>Note:</b> The following workload selection keywords act as filters to limit data collection to a specific part of the IMS workload		
CLASS	class	<p>Qualifies an MWAIT request by the numeric scheduling class(es). Multiple classes can be specified as follows:</p> <p>CLASS ==&gt; 1 2 3</p> <p>Default is to include all IMS scheduling classes.</p> <p><b>Note:</b> This workload selection keyword cannot be used for a request for DBCTL threads.</p>

Table 5. SET Keywords to Activate a Workload Wait (MWAIT) Service (continued)

Keyword	Operand	Description
PGMTYPE  PTYP	MPP MDP IFP FPU TPI BMP DBT NOTDBT	<p>Qualifies a monitor request by program type. Multiple program types can be specified. The types are:</p> <p>MPP      Message processing program and Java message processing program (JMP)</p> <p>MDP      Message-driven Fast Path program</p> <p>IFP      IMS Fast Path program</p> <p>FPU      Fast Path utility</p> <p>TPI      CPI-C driven program</p> <p>BMP      Batch message processing program or Java batch message processing program (JBP)</p> <p>DBT      DBCTL thread (CICS or ODBA)</p> <p>Default is all types. DBT selects only programs that use a DBCTL thread. NOTDBT selects programs that do not use a DBCTL thread.</p>
PROG	program name	<p>Qualifies an MWAIT request by the specified program. Valid entries are 1- to 8-character alphanumeric program names, which can be used with the plus qualifier (+). Multiple program names can be specified; for example:</p> <p>PROG ==&gt; abc cde+</p>
PSB	psb name	<p>Qualifies an MWAIT request by the specified IMS program specification block. Valid entries are 1- to 8-character alphanumeric PSB names, which can be used with the plus qualifier (+). Multiple PSB names can be specified; for example:</p> <p>PSB ==&gt; ab1 c+2</p>
REGION	jobname NOCOMM	<p>jobname      Qualifies an MWAIT request by the specified job name of an IMS region. Valid entries are 1- to 8-character alphanumeric region job names, which can be used with the plus qualifier (+). Multiple region job names can be specified; for example:</p> <p>REGION ==&gt; +ba ace</p> <p>NOCOMM      Allows monitoring of all regions, excluding communications and including queuing events.</p>
RGNID	region number	<p>Qualifies an MWAIT request by the specified region ID. Valid entries are 1- to 3-character numeric region IDs from 1 to nnn where nnn is a valid region number. Multiple region IDs can be specified; for example:</p> <p>RGNID ==&gt; 1 23 200</p>

Table 5. SET Keywords to Activate a Workload Wait (MWAIT) Service (continued)

Keyword	Operand	Description
TERM	lterm	Qualifies an MWAIT request by the name of the IMS LTERM. Valid entries are 1- to 8-character alphanumeric LTERM names, which can be used with the plus qualifier (+). Multiple LTERM names can be specified; for example:  TERM ==> L071 L0+
	SYNCCLOCK	Defines a monitor request to include <i>all</i> transactions from systems synchronized to the clock of the local IMS (in other words, all local transactions, all local MSC transactions, and all nonlocal MSC transactions that arrive from a system with its clock synchronized to the clock of the local IMS).
	MSCCLOCK	Defines a monitor request to include <i>only nonlocal</i> transactions from MSC systems synchronized to the clock of the local IMS.
TRAN	trancode	Qualifies an MWAIT request by the specified IMS or CICS transaction. Valid entries are 1- to 8-character alphanumeric transaction codes, which can be used with the plus qualifier (+). Multiple transaction codes can be specified; for example:  TRAN ==> TR+ PAY1  For a CICS DBCTL transaction, the CICS transaction code is used.  For a Fast Path transaction, the routing code assigned by the DBFHAGU0 exit routine is used.
TRANSTYPE  TTYTYP	<u>ALL</u>  DB2  DLI FP	Qualifies an MWAIT request by transaction type:  ALL      All transaction types DB2      Transactions that access DB2 DLI      Transactions that make DL/I database calls FP      Transactions that make Fast Path database calls  The default is all types, and multiple transaction types can be specified.
USERID	userid	Qualifies an MWAIT request by the name of the IMS sign-on user ID. Valid entries are to 1- to 8-character alphanumeric IMS sign-on user IDs, which can be used with the plus qualifier (+). Multiple user IDs can be specified; for example:  USERID ==> USER1 USER2 USER6

The workload selection keywords can be used to limit the data collection to a specific part of the IMS workload. By specifying a certain resource to be monitored (such as an IMS region), the sampling is automatically limited to those events for that resource. For example, the only flow components that use an IMS region are scheduling, application program, and sync point.

Input and output communication and queuing events are ignored when a REGION parameter is specified.

**REGION=**

You should specify a REGION parameter to limit the amount of data that must be sampled. NOCOMM, a special keyword for the REGION parameter, can be used to exclude communications and include queuing events. No other REGION or RGNID parameters can be used with the NOCOMM keyword.

---

## Stopping MWAIT

An MWAIT request can be stopped by:

- Stopping the request with a Z line command  
Use the Z line command from the Active Timer list application as described in the chapter “Displaying a List of Active Timer Requests” in *Monitors and Traces Reference Manual*.
- Setting a stop time for automatic completion of data collection  
Specify the STOP value (as a timestamp) on the start or modify MWAIT request panel. The collected data remains available for viewing until the request is purged.
- Purging a request from the Active Timer list with the P line command  
Use the P line command from the Active Timer list application as described in the chapter “Displaying a List of Active Timer Requests” in *Monitors and Traces Reference Manual*.
- Purging a service request with a SET request  
Issue a PRG request with SET from the SERV field of any display, a BBPARM member, or an MAINVIEW AutoOPERATOR EXEC (MAINVIEW AutoOPERATOR must be installed); for example:  

```
SERV ==> SET  
PARM ==> PRG=requ d|ALL
```
- Stopping a request with a SET request  
Use the STOP parameter with SET; for example:  

```
SERV ==> SET  
PARM ==> REQ=MWAIT, SAMPLE, START=00: 10: 00, STOP=00: 30: 00, I=200MSEC
```

The request starts at 10 minutes after midnight and stops 30 minutes after midnight.

---

## Storage Requirements

Each active MWAIT request requires approximately 6K of storage in CSA. This storage is GETMAINED when data collection starts and is freed when the MWAIT request is purged.

---

## Defining a Sampling Interval

The more samples that are taken, the more accurate the workload analysis. An interval of less than one second can be defined by using MSEC with the INTERVAL keyword. The sampling interval (INTERVAL) and period (CURPER and HISTORY) should be defined (defaults are specified in BBPARM member BBIISP00) to take at least 2,000 samples in a period.

For example:

```
INTERVAL ==> 200MSEC
CURPER   ==> 00: 10: 00
HISTORY  ==> 00: 30: 00
TRAN     ==> MYTRAN
```

takes 5 samples per second or 3000 samples per current period (CURPER) and 9,000 samples per history period and the samples are limited to those events for transaction MYTRAN. The number of samples taken in a CURRENT or HISTORY period is displayed at the bottom of the DWAIT display.

**Note:** An interval that is too short could consume too much CPU, depending upon the number of IMS resources defined, the CPU speed, and the I/O rate.

---

## Qualifying a Workload MWAIT Request

To limit the flow components monitored but not the resources, a generic operand of + can be specified for an MWAIT keyword to select all resources of that type. For example, REGION=+ monitors all regions, but excludes input and output events. TRAN=+ monitors all events that have an identifiable transaction code associated with them, which eliminates output queuing and output communications.

The + character can be used generically or positionally. As a generic resource name qualifier, it cannot be followed by any other character. As a positional qualifier, it must be repeated for every character to be replaced.

For example:

```
REGION ==> CICS+
```

is a data collection request for workload wait to sample only those regions whose names begin with CICS.

Entering:

```
TRAN ==> A++N+
```

monitors all events that have an identifiable transaction code with an A character in the first position, any two characters before N, and any character following N.

The + qualifier can be used for MWAIT service requests for:

**Resource Selection**

PROG keyword  
PSB keyword  
REGION keyword  
TERM keyword  
TRAN keyword  
USERID keyword

**Resource**

Program names  
PSBs  
Region job names  
LTERM names  
Transaction names  
User IDs

---

## Chapter 23. Requesting Workload Wait Data Display (DWAIT)

As described in Chapter 22, “Requesting Workload Wait Data Collection (MWAIT)” on page 433, data collection must first be activated with an MWAIT request before DWAIT can be requested. Active MWAIT requests, each identified with its specified parameter, are displayed in the Active Timer Requests application (described in the chapter called “Displaying a List of Active Timer Requests” in *Monitors and Traces Reference Manual*). Use the S line command from this application to display the data collected by MWAIT with the DWAIT service. DWAIT can also be requested from a list of analyzer display services (Primary Menu Option 1) or from the service display panel with a SET request (see the “SET Timer Request” chapter in *Monitors and Traces Reference Manual*).

If an identifier is specified with the MWAIT request, it must be used with DWAIT. For example, for the following MWAIT request identified as MON1:

```
PARM      ==> MON1
INTERVAL  ==> 200MSEC
TRAN      ==> PART
```

The same MON1 identifier must be specified if DWAIT is requested from the list of analyzer display services or the service display panel with a SET request.

```
SERV      ==> DWAIT
PARM      ==> MON1
```

The DWAIT service shows the workload wait events that were observed most frequently by the MWAIT service. These events are the main contributors affecting IMS response time for the sampled workloads. The events are classified into the components of transaction flow through IMS (such as application program) to make it simpler to identify where bottlenecks occur.

To provide a measure of workload throughput and performance to relate to the displayed workload wait events, the number of MPP transactions processed and the average response time measured are provided at the bottom of the display. This information is only available if the Event Collector is active.



## Chapter 24. DWAIT - Workload Wait Display

BMC SOFTWARE -----			DISPLAY WORKLOAD WAIT -----			PERFORMANCE MGMT -----		
SERV ==> DWAIT			INPUT 12: 02: 41			INTVL=> 8 LOG=> N TGT==> IMSxxx		
PARM ==> MON1 TRANS=ALL						SCROLL=> N/A		
EVENTS - All Transactions			--- CURRENT PERIOD ---			----- HISTORY -----		
	%	0	50	100%	%	0	40	80%
INPUT COMMUNICATIONS	1.16	.	.	.		2.62	>	.
Sel Dsp - CIOP	(1.16)	.	.	.		(2.62)	>	.
INPUT QUEUE	83.68	----->				63.23	----->	.
All MPR's Busy	(81.91)	----->				(60.86)	----->	.
MPP GU	(1.77)	.	.	.		(2.37)	>	.
SCHEDULING	1.16	.	.	.		2.62	>	.
FETCH I/O Active	(1.16)	.	.	.		(2.62)	>	.
APPLICATION PROGRAM	3.53	>	.	.		7.96	>	.
DL/I-CTL Using CPU	(0.04)	.	.	.		(0.06)	.	.
Appl Using CPU	(3.49)	>	.	.		(7.86)	>	.
SYNC POINT	3.49	>	.	.		7.86	>	.
LWA Wait	(3.49)	>	.	.		(7.86)	>	.
OUTPUT QUEUE	2.33	.	.	.		5.24	>	.
Node Busy	(2.33)	.	.	.		(5.24)	>	.
OUTPUT COMMUNICATIONS	4.65	>	.	.		10.48	-->	.
MFS Pool Failure	(4.65)	>	.	.		(10.48)	-->	.
		+-----+					+-----+	
RESPONSE TIME (SEC)	1.39	79 SAMPLES			1.39	1,039 SAMPLES		
# OF TRANSACTIONS	67	0 MIN/39 SEC			67	8 MIN/39 SEC		

**Description:**

The DWAIT service displays data collected by the MWAIT service. This display shows the workload events that account for the IMS response time. The data displayed is collected by periodic sampling of the system by the MWAIT service.

**Select Code:**

DWAIT

**Parameter:**

The name used with the MWAIT service request.

FLOWCOMP=xx

Specifies IMS transaction processing event component(s), where xx represents any of the following characters. Any combination can be specified. They are presented below in the same sequence that the events occur during IMS transaction processing.

ALL All IMS processing components

COMP All components are to be displayed even if their sampled events are zero

IC Input communications

IQ Input queue

SC Scheduling

AP Application program

SP	Synchronization point
OQ	Output queue
OC	Output communications

The default is to build a full screen display of the events that are the largest contributors to the transaction response time.

#### FORMAT=% | #

Specifies the format in which the statistics are to be presented.

- % Individual events are to be expressed as a percentage of the total events sampled. This parameter is the default and can also be expressed as **FORMAT=PERCENT**.
- # Displays the average count of transactions in each event per sample. The number of events can be calculated by multiplying the count by the number of samples. This calculation can be used to determine the significance of high-percentage spikes and whether enough samples have been taken for accuracy. This parameter can be expressed as **FORMAT=COUNT**.

#### TRANS=ACTIVE | ALL

Specifies which events are to be displayed and used in calculation of percentages.

##### ACTIVE

Only active events are to be displayed or used in the calculations of percentages; queued events are excluded. This is the default.

##### ALL

All monitored events are to be displayed.

#### Field Descriptions:

Each of the fields is shown and described below by display area.

### Area 1

#### EVENTS - All Transactions

	%
INPUT COMMUNICATIONS	1. 16
Sel Dsp - CIOP	(1. 16)
INPUT QUEUE	83. 68
All MPR's Busy	(81. 91)
MPP GU	(1. 77)
SCHEDULING	1. 16
FETCH I/O Active	(1. 16)
APPLICATION PROGRAM	3. 53
DL/I-CTL Using CPU	(0. 04)
Appl Using CPU	(3. 49)
SYNC POINT	3. 49
LWA Wait	(3. 49)
OUTPUT QUEUE	2. 33
Node Busy	(2. 33)
OUTPUT COMMUNICATIONS	4. 65
MFS Pool Failure	(4. 65)
RESPONSE TIME (SEC)	1. 39
# OF TRANSACTIONS	67

This area shows the IMS processing events that occur as transactions flow through the system. You can see the events that contribute to transaction response time and where transactions spend their time during IMS processing. This area also displays the transaction response time and the number of transactions that completed.

The IMS processing event components that are shown include:

INPUT COMMUNICATIONS  
INPUT QUEUE  
SCHEDULING  
APPLICATION PROGRAM  
SYNC POINT  
OUTPUT QUEUE  
OUTPUT COMMUNICATION

Minor IMS processing events are indented two spaces and are in lowercase letters beneath the major (component) processing event. The values measured for minor events during a sampling are enclosed in parentheses. These values make the total value measured for the major event.

If the value is too small to be displayed in two decimal places, **SMALL** is shown. If the numeric value for the **CURRENT PERIOD** (see "Area 2") or **HISTORY** area of the display (see "Area 3") is zero, **ZERO** is shown for the event value.

The other fields in this area of the display are listed below, in alphabetical order.

#### # OF TRANSACTIONS

Number of MPP or JMP transactions that completed for the current **PERIOD** or **HISTORY** sampling.

Number of DBT transactions that satisfy the **MWAIT** request selection criteria for **CICS** regions attached to an IMS control region if **CI CS=YES** or **CI CS=ONLINE** is specified in **BBPARM** member **IMFECP00**. **BMPs** and **JBPs** are excluded from this count.

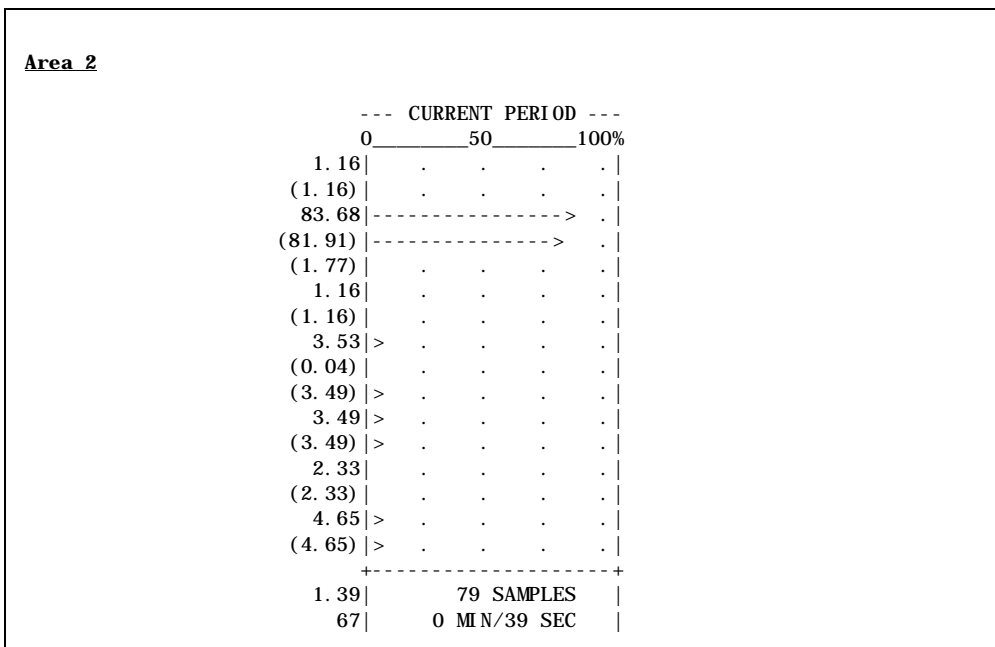
## EVENTS

Identifies the events being displayed as specified by the TRANS parameter.

## RESPONSE TIME

Average response time of the MPP or JMP transactions that completed in the current period or history.

Average response time for CICS regions attached to an IMS control region if CICS=YES or CICS=ONLINE is specified in BBPARM member IMFECP00. BMPs and JBPs are excluded from this count.



This area shows the distribution of the workload for the current period. The CURRENT PERIOD displays the statistics (see “Area 1”) for the last n time intervals as specified by the corresponding MWAIT request. When the CURRENT PERIOD is compared with the HISTORY area of the display (see “Area 3”), response time problems that are transient can be differentiated from ongoing problems.

The descriptions are arranged in alphabetical order.

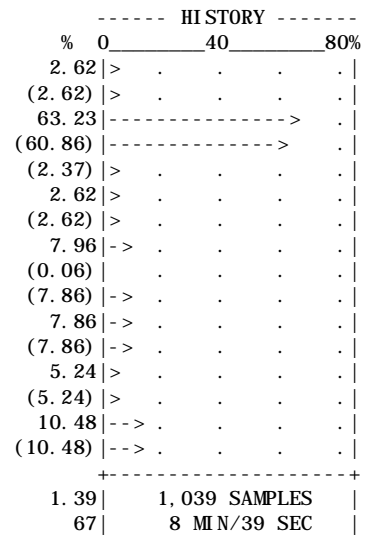
## MIN/SEC

Number of minutes and seconds as defined by the corresponding MWAIT request for the CURRENT PERIOD sampling.

## SAMPLES

Number of samples taken to produce the statistics (see “Area 1”) and the CURRENT PERIOD graph.

**Area 3**



This area shows the distribution of the workload over a long period of time (HISTORY) as specified by the corresponding MWAIT request. When the HISTORY period is compared with the CURRENT PERIOD, response time problems that are ongoing can be differentiated from those that are transient.

The descriptions are arranged in alphabetical order.

**MIN/SEC**  
Number of minutes and seconds as defined by the corresponding MWAIT request for the HISTORY sampling.

**SAMPLES**  
Number of samples taken to produce the statistics for the IMS processing events (see “Area 1”) and HISTORY graph.

---

## Excluding Queued Events

Transactions that are delayed on the IMS input or output queues because of external conditions can have a skewing effect on the DWAIT display. This effect is removed by the DWAIT parameter TRANS=ACTIVE, which is the default. TRANS=ACTIVE excludes all queued events from the display; TRANS=ALL includes all events.

The excluded events are:

### Input Queue

- BMP or JBP not active
- MPP or JMP unschedulable
- BMP or JBP unschedulable
- MPP or JMP class not active

### Output Queue

- LTERM unavailable
- NODE busy
- NODE unavailable
- Line busy
- Line unavailable
- MSC P-link busy
- MSC P-link unavailable
- MSC L-link unavailable

---

## DWAIT Event Definitions

The following sections describe what may be shown for a major IMS processing event component displayed by DWAIT. These descriptions are arranged alphabetically by:

- Application Program (AP)
- Input Communications (IC)
- Input Queue (IQ)
- Output Communications (OC)
- Output Queue (OQ)
- Scheduling (SC)
- Sync Point (SP)

---

## Application Program (AP) Events

The transaction is now at the point where actual application processing is performed. It is at this stage that the majority of workload slowdowns usually occur. This transaction flow component starts at the point IMS branches to the application program in the dependent region and ends when the application program enters sync point.

The following sections list and define the events that may be shown for APPLICATION PROGRAM by the DWAIT display. The descriptions are in alphabetical order.

### AP: Appl CPU Wait

The application program was executing in the dependent region and was not waiting for any event to occur. The program work could be executed but there were no CPUs available to process it. In the case of a UP processor, the CPU was executing other work (IMS or non-IMS) in the system. For MP processors, all the CPUs were busy with other work.

From an OS viewpoint, the dependent region program control task was in the OS dispatcher work-to-do queue. A high occurrence of this event indicates that the CPU is not fast enough or that excessive work of a higher priority was using the CPU cycles.

### AP: Appl CSA Fault

The same as “AP: Appl Prv Fault” on page 452, except that the application program was accessing data or instructions in CSA.

### AP: Appl CSA GFA

The same as “AP: Appl Prv GFA” on page 452, except that the program was accessing data or instructions in CSA.

### AP: Appl I/O Active

During a sampling of the system, the application program had issued an I/O request directly to OS for a non-IMS database file. At the time of the sample, the I/O was active against the file. Processing of the message is suspended until the I/O completes.

### AP: Appl I/O Queued

The same as “AP: Appl I/O Active”, except that the application program I/O was queued on the logical channel or UCB because I/O contention prevented the successful initiation of the SIO instruction.

### AP: Appl LPA Fault

The same as “AP: Appl Prv Fault” on page 452, except that the application program was accessing data or instructions in LPA.

## **AP: Appl LPA GFA**

The same as “AP: Appl Prv GFA”, except that the program was accessing data or instructions in LPA.

## **AP: Appl Prv Fault**

During a sampling of the system, the application program was processing a message and a page fault occurred. Processing of the message is delayed until the fault is resolved.

For an application program, this means that the program in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

## **AP: Appl Prv GFA**

During a sampling of the system, the application program was processing a message and a page fault occurred. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed because no real frames were available. The page fault is placed in the general frame allocation (GFA) queue while the situation is dealt with.

For the application program, this means that the program in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## **AP: Appl Using CPU**

The application program was executing in the dependent region and was not waiting for any event to occur. The program could be executed and, in fact, a CPU was actively processing it. In the case of a UP processor, the CPU was executing this application program. For MP processors, one of the CPUs was executing the application program.

A high occurrence of this event indicates that the application program is looping or consuming lots of CPU for some other reason.

## **AP: Application in CICS**

During a sampling of the system, the application was under control of the CICS address space. A large percentage of time spent under this event could indicate a bottleneck in the CICS address space, but it could be normal if the transaction does more processing in CICS than in DL/I.

## **AP: CBTS Latch**

During a sampling of the system, IMS was processing a DL/I call and required the dynamic control block latch but the latch was in use by another IMS task.

## **AP: CTL SYNC I/O Active**

The same as “SC: CTL SYNC I/O Active” on page 485 except that the application program processing has begun.

## **AP: CTL SYNC I/O Queued**

The same as “SC: CTL SYNC I/O Queued” on page 485, except that the application program processing has begun.

## **AP: DB2N-Dep CPU Wait**

The application program in the dependent region has issued a normal SQL call and is waiting to be dispatched by the operating system. A high occurrence of this event indicates that either the CPU is too slow to process the workload or that the dispatching priority of the dependent region is not high enough and higher priority tasks in the system are being selected for processing.

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

## **AP: DB2N-Dep CSA Fault**

The same as “AP: DB2N-Dep Prv Fault” except that the DB2 interface code was accessing data or instructions in CSA.

## **AP: DB2N-Dep CSA GFA**

The same as “AP: DB2N-Dep Prv GFA” on page 454 except that the DB2 interface code was accessing data or instructions in CSA.

## **AP: DB2N-Dep LPA Fault**

The same as “AP: DB2N-Dep Prv Fault” except that the DB2 interface code was accessing data or instructions in LPA.

## **AP: DB2N-Dep LPA GFA**

The same as “AP: DB2N-Dep Prv GFA” on page 454 except that the DB2 interface code was accessing data or instructions in LPA.

## **AP: DB2N-Dep Prv Fault**

During a sampling of the system, the application program had issued an SQL call and the DB2 interface code had a page fault while accessing data or instructions in the private region during call processing.

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

## AP: DB2N-Dep Prv GFA

During a sampling of the system, the application program had issued an SQL call. The DB2 interface code had suffered a page fault while accessing data or instructions in the private region during call processing. The required page could not be paged in because of a shortage of real storage frames to do the I/O. The page fault is placed on the general frame allocation (GFA) queue until a frame is made available. When a page fault occurs, the system resource manager (SRM) is called to steal a page from existing users. This could result in a page-out operation.

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

## AP: DB2N-Dep Using CPU

The application program in the dependent region issued a normal SQL call and was actively processing the call on a CPU. A high occurrence of this event indicates that DB2 is consuming a large amount of CPU to process the call. This could be because of:

- A complex SQL call
- Inefficient usage of DB2 buffer pools
- Inefficient index access
- Bad DB2 table design
- Entire table scan
- Badly coded SQL statement

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

## AP: DB2N-Dep Wait

The application program in the dependent region issued a normal SQL call and is in a wait for processing by DB2. A high occurrence of this event could indicate resource contention in DB2, such as:

- Insufficient pool space in the buffer pools or EDM pool
- Lock contention
- Table OPEN/CLOSE processing

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

## AP: DB2S-Dep CPU Wait

The same as “AP: DB2N-Dep CPU Wait” on page 453 except that the DB2 interface code was processing a service call.

A service call is issued by the IMS DB2 interface code for certain events in the dependent region. A CREATE THREAD call is issued to identify the transaction to DB2 and a COMMIT PHASE 1 and PHASE 2 call is done at IMS synchronization point time for data integrity.

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

## **AP: DB2S-Dep CSA Fault**

The same as “AP: DB2N-Dep CSA Fault” on page 453 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep CSA GFA**

The same as “AP: DB2N-Dep CSA GFA” on page 453 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep LPA Fault**

The same as “AP: DB2N-Dep LPA Fault” on page 453 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep LPA GFA**

The same as “AP: DB2N-Dep LPA GFA” on page 453 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep Prv Fault**

The same as “AP: DB2N-Dep Prv Fault” on page 453 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep Prv GFA**

The same as “AP: DB2N-Dep Prv GFA” on page 454 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep Using CPU**

The same as “AP: DB2N-Dep Using CPU” on page 454 except that the DB2 interface code was processing a service call.

## **AP: DB2S-Dep Wait**

The same as “AP: DB2N-Dep Wait” on page 454 except that the DB2 interface code was processing a service call.

## **AP: DC Mon Latch**

During a sampling of the system, the application program issued a DL/I call which needed the DC Monitor latch. The latch was already in use by another IMS task. Processing is suspended until the latch becomes available.

The DC Monitor latch is used to serialize the use of the DC Monitor in IMS. DL/I attempts to move data to the DC monitor buffers. Each call to move a new record to the buffers requires this latch. A high occurrence of this event indicates that the DC Monitor is very busy or that the DC Monitor is hung up in its I/O attempt to DASD or the tape file.

## **AP: DEDB Area Latch**

During a sampling of the system, one or more transactions were waiting for a specific DEDB area.

## **AP: DEDB CI Contention**

During a sampling of the system, one or more transactions were in contention for the same DEDB control interval.

## **AP: DEDB Latch**

During a sampling of the system, one or more transactions were waiting for a DEDB latch.

## **AP: DL/I-CTL CPU Wait**

The same as “AP: DL/I-Dep CPU Wait” on page 457, except that DL/I has switched to the control task in the control region.

## **AP: DL/I-CTL CSA Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in CSA while executing under the LSO task in the control region.

## **AP: DL/I-CTL CSA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in CSA while executing under the control task in the control region.

## **AP: DL/I-CTL LPA Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in LPA while executing under the LSO task in the control region.

## **AP: DL/I-CTL LPA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in LPA while executing under the control task in the control region.

## **AP: DL/I-CTL Prv Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in the private area of the control region while executing under the control task in the control region.

## **AP: DL/I-CTL Prv GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in the private area of the control region while executing under the control task in the control region.

## **AP: DL/I-CTL Using CPU**

The same as “AP: DL/I-Dep Using CPU” on page 458, except that DL/I has switched to the control task in the control region.

## **AP: DL/I-Dep CPU Wait**

The application program in the dependent region has issued a DL/I call and DL/I was executing in the dependent region and was not waiting for any event to occur. DL/I could be executed but there were no CPUs available to process it. In the case of a UP processor, the CPU was executing other work (IMS or non-IMS) in the system. For MP processors, all the CPUs were busy with other work.

From an OS viewpoint, the dependent region program control task was in the OS dispatcher work-to-do queue. A high occurrence of this event indicates that the CPU is not fast enough or that excessive work of a higher priority was using the CPU cycles.

## **AP: DL/I-Dep CSA Fault**

The same as “AP: DL/I-Dep Prv Fault”, except that DL/I was accessing data or instructions in CSA.

## **AP: DL/I-Dep CSA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in CSA.

## **AP: DL/I-Dep LPA Fault**

The same as “AP: DL/I-Dep Prv Fault”, except that DL/I was accessing data or instructions in LPA.

## **AP: DL/I-Dep LPA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in LPA.

## **AP: DL/I-Dep Prv Fault**

During a sampling of the system, the application program had issued a DL/I call and DL/I suffered a page fault in the private area of the dependent region. Message is delayed until the fault is resolved.

For a DL/I call, this means that DL/I in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

## **AP: DL/I-Dep Prv GFA**

During a sampling of the system, the application program had issued a DL/I call and DL/I suffered a page fault. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed because no real frames were available. The page fault is placed on the general frame allocation (GFA) queue until the situation is resolved.

For the DL/I call, this means that DL/I in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## **AP: DL/I-Dep Using CPU**

The application program in the dependent region has issued a DL/I call and DL/I was executing in the dependent region and was not waiting for any event to occur. DL/I could be executed and, in fact, a CPU was actively processing it. In the case of a UP processor, the CPU was executing this DL/I call. For MP processors, one of the CPUs was executing the DL/I call.

A high occurrence of this event indicates that the database design is such that DL/I is consuming a large amount of CPU to access the requested data. Examples of DL/I configurations that may result in considerable CPU needs are:

- Logical relationships
- Secondary indices
- Long twin chains
- Insufficient Root Anchor Points (RAPs) in an HDAM database
- Calls that require data set OPENs

## **AP: DL/I-LSO CPU Wait**

The same as “AP: DL/I-Dep CPU Wait” on page 457, except that DL/I has switched to the LSO task in the control region. LSO=Y was specified in the control region JCL.

## **AP: DL/I-LSO CSA Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in CSA while executing under the LSO task in the control region.

## **AP: DL/I-LSO CSA GFA**

The same as “AP: DL/I-Dep Prv GFA”, except that DL/I was accessing data or instructions in CSA while executing under the LSO task in the control region.

## **AP: DL/I-LSO LPA Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in LPA while executing under the LSO task in the control region.

## **AP: DL/I-LSO LPA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in LPA while executing under the LSO task in the control region.

## **AP: DL/I-LSO Prv Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in the private area of the control region while executing in cross-task in the control region.

## **AP: DL/I-LSO Prv GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in the private area of the control region while executing under the LSO task in the control region.

## **AP: DL/I-LSO Using CPU**

The same as “AP: DL/I-Dep Using CPU” on page 458, except that DL/I has switched to the LSO task in the control region. LS0=Y was specified in the control region JCL.

## **AP: DL/I-SAS CPU Wait**

The same as “AP: DL/I-Dep CPU Wait” on page 457, except that DL/I has switched to the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS CSA Fault**

The same as “AP: DL/I-CTL Prv Fault” on page 456, except that DL/I was accessing data or instructions in CSA while executing under the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS CSA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in CSA while executing under the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS LPA Fault**

The same as “AP: DL/I-CTL Prv Fault” on page 456, except that DL/I was accessing data or instructions in LPA while executing under the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS LPA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in LPA while executing under the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS Prv Fault**

The same as “AP: DL/I-CTL Prv Fault” on page 456, except that DL/I was accessing data or instructions in the private area of the DL/I subordinate address space while executing under the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS Prv GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in the private area of the DL/I subordinate address space while executing under the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-SAS Using CPU**

The same as “AP: DL/I-Dep Using CPU” on page 458, except that DL/I has switched to the DL/I serial task in the DL/I subordinate address space.

## **AP: DL/I-XMEM CPU Wait**

The same as “AP: DL/I-Dep CPU Wait” on page 457, except that DL/I was using hardware cross-memory op codes to execute DL/I code in the private area of the control region address space or DL/I subordinate address space. LS0=X or LS0=S was specified in the control region JCL.

## **AP: DL/I-XMEM CSA Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in CSA while executing in cross-memory mode.

## **AP: DL/I-XMEM CSA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in CSA while executing in cross-memory mode.

## **AP: DL/I-XMEM LPA Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in LPA while executing in cross-memory mode.

## **AP: DL/I-XMEM LPA GFA**

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in LPA while executing in cross-memory mode.

## **AP: DL/I-XMEM Prv Fault**

The same as “AP: DL/I-Dep Prv Fault” on page 457, except that DL/I was accessing data or instructions in the private area of the control region or the private area of the DL/I subordinate address space while executing in cross-memory mode.

## AP: DL/I-XMEM Prv GFA

The same as “AP: DL/I-Dep Prv GFA” on page 458, except that DL/I was accessing data or instructions in the private area of the control region or the private area of the DL/I subordinate address space while executing in cross-memory mode.

## AP: DL/I-XMEM Using CPU

The same as “AP: DL/I-Dep Using CPU” on page 458, except that DL/I was using hardware cross-memory op codes to execute DL/I code in the private area of the control region address space or DL/I subordinate address space. LS0=X or LS0=S was specified in the control region JCL.

## AP: FP Resource Latch

During a sampling of the system, it was noted that one or more transactions were waiting for a Fast Path resource that needs to be serialized.

## AP: Generic Latch

During a sampling of the system, DL/I needed a generic latch but the latch was already in use by another IMS task.

A generic latch is one that locks a caller-defined entity. Any part of IMS that requires serialization but is not defined under one of the other specific latches is locked by a generic latch. Generic latches are used to lock updates for the items in the following table:

Table 6. Latch Locks

Generic Latch	Latch Target
DMBE	Data management block (DMB) updates
VSBP	VSAM buffer pool master updates
DBBP	OSAM buffer pool master updates

## AP: IRLM Wait

During a sampling of the system, a program executing in a dependent region issued a DL/I call that required database block level sharing. The database segment needed was already in use by another program in another IMS subsystem or pass-the-buck processing with another IRLM required a wait.

During the processing of DL/I calls against databases, IMS ensures that database integrity is maintained by enqueueing database segments to prevent multiple updates. To affect this enqueue across systems, the IMS resource lock manager (IRLM) is called to enqueue the block or control interval (CI).

A high occurrence of this event indicates that there is considerable contention in IRLM processing. There may be too much block sharing of the database occurring.

## **AP: LGMSG I/O Active**

The same as “IC: LGMSG I/O Active” on page 469, except that the application program has issued a DL/I call against a TP PCB that resulted in I/O to the LGMSG data set.

## **AP: LGMSG I/O Queued**

The same as “IC: LGMSG I/O Queued” on page 470, except that the application program has issued a DL/I call against a TP PCB that resulted in I/O to the LGMSG data set.

## **AP: Log Buffer Wait**

The same as “SC: Log Buffer Wait” on page 493, except that it occurred during application DL/I processing.

## **AP: Logical Log Latch**

The same as “IC: Logical Log Latch” on page 470, except that it is DL/I that needs the latch.

## **AP: LWA Wait**

During a sampling of the system, DL/I processing was in a LOG WRITE AHEAD (LWA) wait status. It was waiting for a database change record to be physically written to the log before the actual database could be updated. This is done to ensure data integrity in case of failure.

If necessary, IMS will write the log buffer containing this record, even if the buffer is not yet full. This is the normal signal for the I/O operation (see “SC: Log Buffer Wait” on page 493 for more information on logging.).

LOG WRITE AHEAD (LWA) is also referred to as LOG TAPE WRITE AHEAD (LTWA). LOG WRITE AHEAD calls generally occur at sync point, which is when most physical database updates occur. They may also occur during application processing if IMS needs to use a database buffer that contains changed database records to access a different database block (buffer steal). The number of buffer steals may be reduced by increasing the number of buffers in these database subpools.

DB LWA is a required function. The DCLWA keyword in the IMS SYSGEN is used to activate or deactivate DC LWA.

## **AP: Misc DLI-CTL Wait**

The same as “AP: Misc DLI-Dep Wait”, except that DL/I is waiting under the control task in the control region.

## **AP: Misc DLI-Dep Wait**

This event is detected when the sampler finds DL/I was executing in the dependent region but was waiting in the IMS dispatcher for an event to complete. The sampler could not attribute the wait to any other category in the application program component. An IWAIT was issued by DL/I while executing in the dependent region.

If this event is ever a significant contributor to response time, call BMC Software Customer Support.

## **AP: Misc DLI-LSO Wait**

The same as “AP: Misc DLI-Dep Wait”, except that DL/I is waiting under the LSO task in the control region.

## **AP: Misc DLI-SAS Wait**

The same as “AP: Misc DLI-Dep Wait”, except that DL/I is waiting under the DL/I serial task in the DL/I subordinate address space.

## **AP: Misc DLI-XMEM Wait**

The same as “AP: Misc DLI-Dep Wait”, except that DL/I is waiting while in cross-memory mode using the hardware cross-memory op codes.

## **AP: MSDB Latch**

During a sampling of the system, one or more transactions were waiting for an MSDB latch.

## **AP: OS Wait DL/I-CTL**

The same as “AP: OS Wait DL/I-Dep”, except that DL/I had switched to the CTL task and then went into the OS wait.

This is very serious because the control task supports the entire TP network; the network is hung while OS processes this DL/I request.

The following DL/I processing is always performed under the control task in the control region:

- DB OPEN, CLOSE, and EOV
- HSAM access
- CA Splits
- CI Splits
- VSAM BGWRT
- TP PCB calls
- VSAM database access if LSR pools

If this event is observed with high values, notify BMC Software Customer Support because DL/I OS requests generally are reported under other events within this flow component.

## **AP: OS Wait DL/I-Dep**

During a sampling of the system, the application program had issued a DL/I call. DL/I was processing the call in the dependent region (through parallel DL/I) and DL/I had issued an OS request (macro, SVC, and the like), which resulted in an OS wait (WAIT).

If this event is observed with high values, notify BMC Software Customer Support because IMS OS requests generally are reported under other events within this flow component.

## **AP: OS Wait DL/I-LSO**

The same as “AP: OS Wait DL/I-Dep”, except that DL/I had switched to the LSO task and then went into the OS wait. LSO=Y was specified in the control region JCL.

If this event is observed with high values, notify BMC Software Customer Support because DL/I OS requests generally are reported under other events within this flow component.

## **AP: OS Wait DL/I-SAS**

The same as “AP: OS Wait DL/I-Dep”, except that DL/I had switched to the DL/I serial task in the DL/I subordinate address space and then went into the OS wait.

If this event is observed with high values, notify BMC Software Customer Support because DL/I OS requests generally are reported under other events within this flow component.

## **AP: OS Wait DL/I-XMEM**

The same as “AP: OS Wait DL/I-Dep”, except that DL/I was executing in the dependent region address space, but executing modules reside in the private area of the control region or DL/I subordinate address space. This is done by using cross-memory hardware op codes.

If this event is observed with high values, notify BMC Software Customer Support because DL/I OS requests generally are reported under other events within this flow component.

## **AP: OS Wait In Appl**

During a sampling of the system, the application program was processing a message and had issued an OS request (macro, SVC, for example), which resulted in an OS wait (WAIT).

If this event is observed with high values, the application programs are being significantly degraded by OS requests outside the control of IMS.

## **AP: OSAM DB I/O Active**

During a sampling of the system, the application program had issued a DL/I call that resulted in an OSAM I/O. At the time of the sample, the I/O was active against the database. Processing of the message is suspended until the I/O completes.

## **AP: OSAM DB I/O Queued**

The same as “AP: OSAM DB I/O Active”, except that the OSAM database I/O was queued on the logical channel or UCB because I/O contention prevented the successful initiation of the SIO instruction.

## **AP: Other Latch**

During a sampling of the system, DL/I needed a specific IMS latch but the latch was already in use by another IMS task.

The data collector could not determine which latch was needed. Call BMC Software Customer Support if this event occurs frequently.

## AP: Overflow Buffer Latch

A transaction is waiting for the overflow buffer latch. A high percentage of this event requires investigation of the OBA and NBA parameters in the regions. By increasing the number of normal buffers (NBA), the overflow buffers may not be used and the wait for the latch will not occur. Only one region can use the overflow buffers at one time.

## AP: PI ENQ/DEQ Latch

During a sampling of the system, a program executing in a dependent region had issued a DL/I call that required the enqueueing of a database segment. The latch used to serialize the IMS routines that enqueue segments was already in use; this region had to wait.

During the processing of DL/I calls against databases, IMS ensures that database integrity is maintained by enqueueing database segments to prevent multiple updaters. This is called program isolation (PI) and is one of the mechanisms used by IMS to achieve its high level of data integrity.

This event is not the same as a PI wait. A PI wait occurs when an application program requests access to data that is already in use by another program. A PI latch wait occurs when an application program requests access to data and DL/I must check the enqueue status of the data, but the enqueue routine is already in use by another program.

A high occurrence of this event indicates that IMS is processing a very large number of PI enqueues and dequeues, or many page faults are occurring while it is accessing the PI pool.

## AP: PI Wait

During a sampling of the system, a program executing in a dependent region had issued a DL/I call that required the enqueueing of a database segment. The database segment needed was already in use by another program and so this region had to wait.

During the processing of DL/I calls against databases, IMS ensures that database integrity is maintained by enqueueing database segments to prevent multiple updaters. This is called program isolation (PI) and is one of the mechanisms used by IMS to achieve its high level of data integrity.

A high occurrence of this event indicates that there is considerable contention for some database segments. The use of PROCOPT=E should be considered, as well as a redesign of the database to relieve the contention.

## AP: QBLKS I/O Active

The same as “IC: QBLKS I/O Active” on page 474 except that the application program has issued a DL/I call against a TP PCB that resulted in I/O to the QBLKS data set.

## AP: QBLKS I/O Queued

The same as “IC: QBLKS I/O Queued” on page 474, except that the application program has issued a DL/I call against a TP PCB that resulted in I/O to the QBLKS data set.

## **AP: SAS SYNC I/O Active**

The same as “SC: SAS SYNC I/O Active” on page 497, except that the application program processing has begun.

## **AP: SAS SYNC I/O Queued**

The same as “SC: SAS SYNC I/O Queued” on page 497, except that the application program processing has begun.

## **AP: SHMSG I/O Active**

The same as “IC: SHMSG I/O Active” on page 475, except that the application program has issued a DL/I call against a TP PCB that resulted in I/O to the SHMSG data set.

## **AP: SHMSG I/O Queued**

The same as “IC: SHMSG I/O Queued” on page 476, except that the application program has issued a DL/I call against a TP PCB that resulted in I/O to the SHMSG data set.

## **AP: Storage Mgt Latch**

The same as “IC: Storage Mgt Latch” on page 476, except that it is DL/I that needs the latch.

## **AP: VSAM DB I/O Active**

During a sampling of the system, the application program had issued a DL/I call that resulted in a VSAM I/O. At the time of the sample, the I/O was active against the database. The processing of the message is suspended until the I/O completes.

## **AP: VSAM DB I/O Queued**

The same as “AP: VSAM DB I/O Active”, except that the VSAM database I/O was queued on the logical channel or UCB because I/O contention prevented the successful initiation of the SIO instruction.

## **AP: Waiting for Buffer**

During a sampling of the system, one or more transactions were waiting for a Fast Path buffer.

## **AP: Waiting for CTL Task**

The application program in the dependent region has issued a DL/I call, such as a message call, that requires processing under the control task in the control region.

The work is ready to be executed, but the control task is currently processing a different task so that this region's work must wait for dispatching by IMS.

## **AP: Waiting for SAS Task**

The application program in the dependent region has issued a DL/I call, such as a message call, that requires processing under the DL/I serial task in the DL/I subordinate address space.

The work is ready to be executed, but the DL/I subordinate address space is currently processing a different task so that this region's work must wait for dispatching by IMS.

---

## Input Communications (IC) Events

When a transaction enters the host from the network, it is first processed by input communications. This processing is a combination of an OS/390 access method and IMS formatting. Basically, input communications is responsible for receiving a message from the terminal network and placing it in the IMS input queue.

The following subsections list and define the events that may be shown for INPUT COMMUNICATIONS by the DWAIT display. They are arranged in alphabetical order.

### IC: CBTS Latch

During a sampling of the system, IMS was processing a message and required the dynamic control block latch, but the latch was in use by another IMS task.

During a sampling of the system, while an input message was processing, additional space was needed in the communications input/output pool (CIOP), but sufficient free space was not available to meet the requirement. The CIOP is used by IMS for buffering network traffic and as a work area for editing MFS messages. An area is taken at control region startup for use by the MFS TEST facility (MFS= on the control region JCL) and VTAM RECEIVE ANY buffers are taken out of this pool at control region startup. The amount of CIOP used for RECEIVE ANY buffers is equal to the size of one buffer times the number requested (both on COMM macro in SYSDEF).

When this event occurs, the entire BTAM line or VTAM node is suspended until enough CIOP space becomes available. Consequently, this can be a very serious source of response network stoppage.

If this event is observed, it usually means that the TPDP= parameter on the control region JCL must be increased. Care must be taken not to specify too large a value because this area will be statically GETMAINed in the control region private address space.

### IC: CPU Wait

During a sampling of the system, IMS was processing a message and was ready to use the CPU, but all CPUs were busy on higher priority work. Processing of the message is delayed until a CPU is available.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task, was not receiving CPU cycles when needed.

### IC: CSA Fault

During a sampling of the system, IMS was processing a message and suffered a page fault. Processing of the message is delayed until the fault is resolved.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task, was accessing data or instructions in CSA when the fault occurred.

## IC: CSA GFA CSA

During a sampling of the system, IMS was processing a message and a page fault occurred. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed, because no real frames were available. The page fault is placed on the general frame allocation (GFA) queue until the situation is resolved.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task, was accessing data or instructions in CSA space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## IC: DC Mon Latch

During a sampling of the system, IMS was processing a message and needed the DC Monitor latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The DC monitor latch serializes the use of the DC Monitor in IMS. Whenever the DC Monitor is active, each call for it to move a new record to the buffers requires this latch. A high occurrence of this event indicates that the DC Monitor is very busy or that the DC Monitor is hung up in its attempt to do I/O to the DASD or tape file.

## IC: Generic Latch

During a sampling of the system, IMS was processing a message and needed a generic latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

A generic latch is one that locks a caller-defined entity. Any part of IMS that requires serialization, but is not defined under one of the other specific latches, is locked by a generic latch. Currently, generic latches are used to lock updates for the items in the following table.

Table 7. Latch Locks

Generic Latch	Latch Target
DMBE	Data management block (DMB) updates
VSBP	VSAM buffer pool master updates
DBBP	OSAM buffer pool master updates

## IC: LGMSG I/O Active

During a sampling of the system, IMS was processing a message and had I/O active against the LGMSG data set. Processing of the message is suspended until the I/O completes.

The LGMSG data set is used to save all message traffic for those messages IMS determines to be large. Excessive values for this event indicate that the queue buffer pool size should be

increased (QBUF= on the control region JCL). If the I/O to SHMSG is not approximately equal to the I/O to LGMSG, respecify the LRECL and BLKSIZE of both to balance the I/O to both data sets.

## **IC: LGMSG I/O Queued**

During a sampling of the system, IMS was processing a message and had I/O queued for the LGMSG data set. Processing of the message is suspended until the I/O completes.

The LGMSG data set is used to save all message traffic for those messages IMS determines to be large. Excessive values for this event indicate that contention exists for the channel, control unit, head of string, or spindle where LGMSG is allocated. This could be caused by over-use by this host or by contention (RESERVE, for example) with another host.

## **IC: Log Buffer Wait**

During a sampling, it was found that a buffer was not available for the current record to be logged and the communications task that created the record was waiting until a buffer was written to the data set.

Every major event that occurs in IMS results in a record being written to the IMS log. From this log, IMS supports recovery and accounting. This log is usually on tape to handle the volume produced.

Because of the critical nature of this data, the logging routines cause any task requesting their services (such as writing a record describing an event) to wait until the logging can be completed. Records are posted to an in-storage buffer, which is usually written to the log only when full. While one buffer is being written, posting can continue to another buffer, if available.

If many log buffer waits are occurring, the buffer size and/or number of buffers should be increased. Contention in the logging function can have a critical performance impact.

## **IC: Logical Log Latch**

During a sampling of the system, IMS was processing a message and needed the logical logger latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The logical logger latch serializes the use of the logical logger in IMS. The logical logger is that part of IMS responsible for maintaining the log buffers before they are written to IEFRDER and IEFRDER2. Originally this latch was used for only this purpose; however, in recent releases, this latch is used anytime part of IMS wants to serialize the entire online system. Consequently, a high occurrence of logical latch waits is generally a sign that something is really wrong outside of IMS. For example, if the log tape drive is not ready, the first sign of this in IMS is that the log buffers are all full and each task, as it attempts to place its log record in a log buffer, has to wait for this latch.

## **IC: LPA Fault**

During a sampling of the system, IMS was processing a message and suffered a page fault. Processing of the message is delayed until the fault is resolved.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task, was accessing data or instructions in LPA when the fault occurred.

## **IC: LPA GFA**

During a sampling of the system, IMS was processing a message and suffered a page fault. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed because no real frames were available. The page fault is placed on the general frame allocation (GFA) queue until the situation is resolved.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task was accessing data or instructions in LPA space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## **IC: LWA Wait**

During a sampling of the system, IMS input message processing was in a DC LOG WRITE AHEAD (LWA) wait status. It was waiting for a log record to be physically written to the log before processing could continue. This ensures data integrity in case of failure.

If necessary, IMS writes the log buffer containing this record even if the buffer is not yet full. This is the normal signal for the I/O operation (see “SC: Log Buffer Wait” on page 493 for more information about logging.).

DB LWA is a required function. The DCLWA keyword in the IMS SYSGEN is used to activate or deactivate DC LWA.

## **IC: MFS Blk I/O Active**

During a sampling of the system, IMS was processing a message and had I/O active against the MFS data set. Processing of the message is suspended until the I/O completes.

The MFS data set contains the MFS control blocks (DIF, MID, MOD, and DOF) needed to edit a message. Excessive values for this event indicate that the MFS pool should be enlarged.

## **IC: MFS Blk I/O Queued**

During a sampling of the system, IMS was processing a message and had I/O queued for the MFS data set. Processing of the message is suspended until the I/O completes.

The MFS data set contains the MFS control blocks (DIF, MID, MOD, and DOF) needed to edit a message. Excessive values for this event indicate that contention exists for the channel, control unit, head of string, or spindle where the MFS data set is allocated. This could be caused by over-use by this host or by contention (RESERVE, for example) with another host.

## IC: MFS Dir I/O Active

During a sampling of the system, IMS was processing a message and had I/O active against the MFS data set directory. Processing of the message is suspended until the I/O completes.

The MFS directory is searched any time an MFS block is needed that is not in the MFS pool and there is no in-core directory entry for it. Excessive values for this event indicate that the incore directory should be expanded or that the MFS pool should be enlarged.

## IC: MFS Dir I/O Queued

During a sampling of the system, IMS was processing a message and had I/O queued for the MFS data set directory. Processing of the message is suspended until the I/O completes.

The MFS directory is searched any time an MFS block is needed that is not in the MFS pool and there is no in-core directory entry for it. Excessive values for this event indicate that contention exists for the channel, control unit, head of string, or spindle where the MFS data set is allocated. This could be caused by over-use by this host or by contention (RESERVE, for example) with another host.

## IC: MFS FRE Unavailable

During a sampling of the system, MFS was in the process of editing a message, but there were no fetch request elements (FREs) available. FREs are used by MFS to manage control blocks in the MFS pool. Each DIF, MID, MOD, and DOF in the pool, and each one that is read in require one FRE each.

When this event occurs, the entire BTAM line or VTAM node is suspended until a FRE is freed. Consequently, this can be a very serious source of response network stoppage.

If this event is observed, it usually means that the FRE parameter on the control region JCL must be increased. Care must be taken not to specify too large a value because each FRE takes 40 bytes away from the MFS pool.

## IC: MFS Pool Failure

During a sampling of the system, while a message was processing, additional space was needed in the message format services pool (MFSP), but there was not enough continuous free space to meet the requirement. The MFSP is used by IMS to hold MFS control blocks (DIF/MID on input, MOD/DOF on output). In addition, the in-core directory (\$\$IMSDIR) and MFS fetch request elements (FREs) are taken out of the pool at control region startup.

When this event occurs, the entire BTAM line or VTAM node is suspended until enough MFS pool space becomes available. Consequently, this can be a very serious source of response network stoppage.

If this event is observed, it usually means that the FBP parameter on the control region JCL must be increased. Care must be taken not to specify too large a value because this area will be statically GETMAINED in the control region private address space. The amount of the MFS pool space used for FREs is equal to the number of FREs (FRE= on control region JCL) times 40 bytes, the length of 1 FRE. The size of \$\$IMSDIR can be determined by running the MFS service utility with a LIST INDEX or by calculating its size, knowing that each directory entry is 14 bytes long. This information is also available from the MFSST service.

## IC: OS Wait

During a sampling of the system, IMS was processing a message and had issued an OS request (macro or SVC, for example), which resulted in an OS wait (WAIT). The data collector could not assign the wait to any other category in input or output communications.

If this event is ever observed with high values, please contact BMC Software Customer Support.

## IC: Other IMS Wait

During a sampling of the system, IMS was processing a message and had issued an internal IMS wait (IWAIT). The data collector was unable to assign the wait to any other category in input or output communications.

If this event is ever observed with high values, please contact BMC Software Customer Support.

## IC: Other Latch

During a sampling of the system, IMS was processing a message and needed a specific IMS latch, but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The data collector could not determine which latch was needed. Call BMC Software Customer Support if this event occurs.

## IC: PI ENQ/DEQ Latch

During a sampling of the system, IMS was processing an input message that required executing an enqueue of a database segment. The latch used to serialize the IMS routines that enqueue segments was already in use; this message had to wait.

During the processing of calls against databases, IMS ensures that database integrity is maintained by enqueueing database segments to prevent multiple updating. This is called program isolation (PI) and is one of the mechanisms used by IMS to achieve its high level of data integrity.

Do not confuse this event with a PI wait. A PI wait occurs when an application program requests access to data that is already in use by another program. A PI latch wait occurs when an application program requests access to data and DL/I must check the enqueue status of the data, but the enqueue routine is already in use by another program.

A high occurrence of this event indicates that IMS is processing a very large number of PI enqueues and dequeues, or many page faults are occurring while it is accessing the PI pool.

## IC: Prv Fault

During a sampling of the system, IMS was processing a message and a page fault occurred. Processing of the message is delayed until the fault is resolved.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task, was accessing data or instructions in the private area of the control region address space when the fault occurred.

## **IC: Prv GFA**

During a sampling of the system, IMS was processing a message and a page fault occurred. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed because no real frames were available. The page fault is placed on the general frame allocation (GFA) queue until the situation is resolved.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task was accessing data or instructions in the private control region address space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## **IC: QBLKS I/O Active**

During a sampling of the system, IMS was processing a message and had I/O active against the QBLKS data set. Processing of the message is suspended until the I/O completes.

The QBLKS data set is used to anchor various messages found in SHMSG and LGMSG. Excessive values for this event indicate that the queue buffer pool should be increased (QBUF= on the control region JCL).

## **IC: QBLKS I/O Queued**

During a sampling of the system, IMS was processing a message and had I/O queued for the QBLKS data set. Processing of the message is suspended until the I/O completes.

The QBLKS data set is used to anchor various messages found in SHMSG and LGMSG. Excessive values for this event indicate that contention exists for the channel, control unit, head of string, or spindle where QBLKS is allocated. This could be caused by over-use by this host or by contention (RESERVE, and the like) with another host.

## **IC: SAP Unavailable**

During a sampling of the system, an input message was received from the network, but IMS had not dispatched input communications to process the message because no save area prefix (SAP) was available. A SAP is the master control block needed by IMS to dispatch any unit of IMS work. The entire BTAM line or VTAM node is suspended until a SAP is freed by another unit of work. Consequently, this can be a very serious source of response network stoppage.

If this event is observed, it usually means that the SAV parameter on the control region JCL must be increased. Care must be taken not to specify too large a value because each SAP requires approximately 1000 bytes in the control region private address space.

During a sampling of the system, an input message was received from the network, but IMS had not dispatched input communications to process the message because CIOP or HIOP utilization had exceeded 95%, thus preventing further dispatching of new units of IMS communications work. This curtailing of dispatching is called selective dispatching and can be caused by CIOP, HIOP, or SAP shortages. The CIOP is used by IMS for buffering network traffic and as a work area for editing MFS messages. In addition, an area is taken out at control region startup for use by the MFS TEST facility (MFS= in control region JCL) and VTAM RECEIVE ANY buffers are taken out of this pool at control region startup. The amount of CIOP used for RECEIVE ANY buffers is equal to the size of one buffer times the number requested (both on COMM macro in SYSDEF).

When this event occurs, the entire BTAM line or VTAM node is suspended until CIOP utilization drops below 85%. Consequently, this can be a very serious source of response network stoppage.

If this event is observed, it usually means that the TPDP parameter on the control region JCL must be increased. Care must be taken not to specify too large a value because this area will be statically GETMAINed in the control region private address space.

## **IC: Sel Dsp - SAP**

During a sampling of the system, an input message was received from the network, but IMS had not dispatched input communications to process the message because the number of save area prefixes (SAPs) in use exceeded 95%, thus preventing further dispatching of new units of IMS communications work. This curtailing of dispatching is called selective dispatching and can be caused by CIOP or SAP shortages. A SAP is the master control block needed by IMS to dispatch any unit of IMS communications work.

When this event occurs, the entire BTAM line or VTAM node is suspended until SAP utilization drops below 85%. Consequently, this can be a very serious source of response network stoppage.

If this event is observed, it usually means that the SAV parameter on the control region JCL must be increased. Care must be taken not to specify too large a value because each SAP requires 1000 bytes in the control region private address space.

## **IC: SHMSG I/O Active**

During a sampling of the system, IMS was processing a message and had I/O active against the SHMSG data set. Processing of the message is suspended until the I/O completes.

The SHMSG data set is used to save all message traffic for those messages IMS determines to be small. Excessive values for this event indicate that the queue buffer pool should be increased (QBUF= on control region JCL). If the I/O to SHMSG is not approximately equal to the I/O to LGMSG, respecify the LRECL and BLKSIZE of both to balance the I/O to these data sets.

## IC: SHMSG I/O Queued

During a sampling of the system, IMS was processing a message and had I/O queued for the SHMSG data set. Processing of the message is suspended until the I/O completes.

The SHMSG data set is used to save all message traffic for those messages IMS determines to be small. Excessive values for this event indicate that contention exists for the channel, control unit, head of string, or spindle where SHMSG is allocated. This could be caused by overuse by this host or by contention (RESERVE, for example) with another host.

## IC: SPA I/O Active

During a sampling of the system, IMS was processing a message and had I/O active against the scratch pad area (SPA) data set. Processing of the message is suspended until the I/O completes.

The SPA data set is used to hold all SPAs defined to reside on DASD. Excessive values for this event indicate that consideration should be given to converting some of the DASD-based conversations to main storage (CORE).

## IC: SPA I/O Queued

During a sampling of the system, IMS was processing a message and had I/O queued for the SPA data set. Processing of the message is suspended until the I/O completes.

The scratch pad area (SPA) data set is used to hold all SPAs defined to reside on DASD. Excessive values for this event indicate that contention exists for the channel, control unit, head of string, or spindle where SPA is allocated. This could be caused by over-use by this host or by contention (RESERVE, for example) with another host.

## IC: Storage Mgt Latch

During a sampling of the system, IMS was processing a message and needed the storage management latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The storage management latch is used by the main storage management routines in IMS to create and destroy buffer pools and to get and free buffers within those pools. Because of the design of this part of IMS, storage management must be serialized; thus, this latch.

A high occurrence of this event is a sign that IMS is in extreme trouble because storage management executes quite quickly for each request and usually does not suspend. Any requests for storage that require a wait (IWAIT) frees the latch before waiting; therefore, pool space failures does not result in a high occurrence of this latch event.

## IC: Using CPU

During a sampling of the system, IMS was processing a message and a CPU was actively executing instructions on behalf of the task. If a task is not waiting for any of the previously stated reasons, it must be in this state.

For input and output communications, this indicates that a particular communications task (ITASK) in the control region address space under the control task, was receiving CPU cycles.

---

## Input Queue (IQ) Events

The final result of the input communications process is an input message queued on the IMS input queue. It will remain there until a message region or batch message region becomes available to process it. There are several reasons why the IMS message scheduler might be unable to schedule a dependent region with the necessary application program.

Whenever the workload wait sampler accumulates statistics for the target IMS system, each message (subject to user parameters) on the input queue will be accounted for under one of the categories listed in this subsection. These reasons explain the cause of the work slowdown in this area of IMS processing.

The following subsections list and define the events that may be shown for INPUT QUEUE by the DWAIT display. The descriptions are in alphabetical order.

### IQ: All MPR's Busy

During a sampling of the system, there were messages waiting in the input queue for an MPP or JMP that was not active in any region and all the regions capable of processing the associated class were busy with other MPPs or JMPs.

A high occurrence of this event could indicate that the MPPs or JMPs should be reclassified or additional regions should be started.

### IQ: BMP GU

During a sampling of the system, there were messages waiting in the input queue destined for a BMP or JBP that was currently scheduled in one or more regions. The PROCLIM was such that these messages would be processed in one of the same regions that is already active without a reschedule.

### IQ: BMP Not Active

During a sampling of the system, there were messages waiting in the input queue destined for a BMP or JBP program. The BMP or JBP was not scheduled and the messages will remain queued until the BMP or JBP is initiated.

### IQ: BMP Unschedulable

During a sampling of the system, there were messages waiting in the input queue destined for a BMP or JBP, but the BMP or JBP could not be scheduled because of resource problems.

The following resource problems are typical:

- Transaction code (SMB) stopped
- Transaction code (SMB) pstopped
- Transaction code (SMB) locked
- PSB locked
- Program stopped
- PSB-sensitive database stopped
- PSB-sensitive database locked
- PSB not found in ACBLIB
- Program locked

## **IQ: IFP GU**

A transaction is waiting on the balancing group queue for an available IFP. A high percentage of this event indicates that more IFP regions may be required to process Fast Path transactions or the IFP regions cannot process transactions quickly because of some resource contention.

## **IQ: MPP Class Not Active**

During a sampling of the system, there were messages waiting in the input queue destined for an MPP or JMP, but the class of the transaction was not processable by any of the active message regions. The messages remain in the queue until the classes of existing regions are changed or until additional regions are started.

## **IQ: MPP GU**

During a sampling of the system, there were messages waiting in the input queue destined for an MPP or JMP that was currently scheduled in one or more regions. The PROCLIM was such that these messages would be processed in one of the same regions that is already active without a reschedule.

## **IQ: MPP Schd Blocked**

During a sampling of the system, there were messages waiting in the input queue destined for an MPP or JMP, but the MPP or JMP was not scheduled. At least one region with the appropriate class was inactive, but it could not schedule this transaction because the scheduling options (SCHD= on TRANSACT macro) prevented it. The inactive region(s) were suspended because of INTENT or pool space failures, but IMS was not allowed to look for other work to schedule in the region. If it were, this transaction would be scheduled.

A high incidence of this event indicates over-restrictive use of a SCHD parameter in the IMS SYSGEN.

## **IQ: MPP Unschedulable**

During a sampling of the system, there were messages waiting in the input queue destined for an MPP or JMP, but the MPP or JMP could not be scheduled because of resource problems.

The following resource problems are typical:

- Transaction code (SMB) stopped
- Transaction code (SMB) pstopped
- Transaction code (SMB) locked
- PSB locked
- Program stopped
- PSB-sensitive database stopped
- PSB-sensitive database locked
- PSB not found in ACBLIB
- Program locked
- Transaction class stopped
- Transaction class set to zero

## **IQ: PROCLIM Reschedule**

During a sampling of the system, there were messages waiting in the input queue destined for an MPP or JMP that was active in one or more regions, but the PROCLIM value was such that a reschedule had to occur before the queued messages could be processed. |

A high occurrence of this event could be an indication that the PROCLIM values should be increased.

---

## Output Communications (OC) Events

During SYNC point processing all output messages generated by the application program are placed in the output queue. When the destination BTAM line or VTAM node becomes available, output communications is invoked by IMS to send the queued messages. Output communications is a combination of an OS access method and IMS formatting. Basically, output communications is responsible for taking a message out of the queue and sending it to the network.

The events associated with output communications are identical to those for input communications (see “Input Communications (IC) Events” on page 468 for the detailed event descriptions).

---

## Output Queue (OQ) Events

The output messages generated by the application program are made available for transmission as part of SYNC point processing. An exception to this is the messages placed in the queue using ISRT and PURG calls against an EXPRESS TP PCB. These messages are available for transmission immediately following the completion of the PURG DL/I call.

Output messages remain on the IMS output queue until IMS output communications is ready to issue the necessary OS access method request. If the network resources are unavailable, these messages also remain on the queue.

For this major processing event component, the workload wait display provides the information needed to determine if there are any significant delays in output traffic. The reasons why the messages are remaining in the queue are given.

For those messages destined for a remote system, MSC delays caused by link problems are accounted for separately.

The following subsections list and define the events that may be shown for OUTPUT QUEUE by the DWAIT display. The descriptions are in alphabetical order.

### OQ: Line Busy

During a sampling of the system, there was a message in the output queue destined for an LTERM, but the associated BTAM line was busy.

This event may be reported in a system with a VTAM network, if the OS console WTOR is used to communicate with IMS. IMS and MVIMS both consider the WTOR as LINE 1, PTERM 1, and not as a node.

### OQ: Line Unavailable

During a sampling of the system, there was a message in the output queue destined for an LTERM, but the associated BTAM line was unavailable.

This event may be reported in a system with a VTAM network, if the OS console WTOR is used to communicate with IMS. IMS and MVIMS both consider the WTOR as LINE 1, PTERM 1, and not as a node.

The following resource problems are typical:

- BTAM Line stopped
- BTAM Line pstopped
- BTAM Line locked
- BTAM terminal inoperable
- BTAM terminal stopped
- BTAM terminal pstopped
- BTAM terminal locked

## **OQ: LTERM Unavailable**

During a sampling of the system, there was a message in the output queue destined for an LTERM, but the LTERM was unavailable.

The following resource problems are typical:

- LTERM stopped
- LTERM pstopped
- LTERM locked

## **OQ: MSC L-Link Unavailable**

During a sampling of the system, there was a message in the output queue destined for a remote terminal, but the associated MSC logical link was unavailable.

The following resource problems are typical:

- Link stopped
- Link pstopped
- Link locked
- PTERM stopped
- PTERM pstopped
- PTERM locked

## **OQ: MSC P-Link Busy**

During a sampling of the system, there was a message in the output queue destined for a remote terminal, but the associated MSC physical link was busy.

## **OQ: MSC P-Link Unavailable**

During a sampling of the system, there was a message in the output queue destined for a remote terminal, but the associated MSC physical link was unavailable.

The following resource problems are typical:

- Link stopped
- Link pstopped
- Link locked
- Link inoperable
- PTERM stopped
- PTERM pstopped
- PTERM locked

## **OQ: Node Busy**

During a sampling of the system, there was a message in the output queue destined for an LTERM, but the associated VTAM node was busy.

## OQ: Node Unavailable

During a sampling of the system, there was a message in the output queue destined for an LTERM, but the associated VTAM node was unavailable.

The following resource problems are typical:

- VTAM node stopped
- VTAM node pstopped
- VTAM node locked
- VTAM terminal stopped
- VTAM terminal pstopped

---

## Scheduling (SC) Events

When a dependent region becomes available to process an input message, the IMS scheduler is invoked.

The IMS scheduler is responsible for acquiring the resources needed by the application and then invoking the application.

There is a delay between the time the scheduler takes a message out of the queue and the time the application requests it for processing. The workload wait sampler accumulates slowdowns in this area under the scheduler component of flow.

To Workload Analyzer, the actual searching for and loading of the application program in the dependent region is included as scheduler processing. This is a different approach from that taken by other IMS performance tools such as the DC Monitor. Workload Analyzer takes this approach because the activity that occurs in the dependent region seems to be scheduling-type activity and the application program is not responsible for delays in this area. With this approach, the workload wait analyzer differentiates between scheduling work performed in the control region and scheduling work performed in the dependent region, which makes it easy to determine if there is a scheduling problem in the control or dependent region.

The following subsections list and define the events that may be shown for SCHEDULING by the DWAIT display.

### SC: BLR Busy - BMP

The IMS initialization routine for a BMP or JBP region was attempting to allocate the needed PSB and DMBs. Before the BMP or JBP program can be called, the associated PSB and DMBs must be in the main storage pools. The IMS block loader is used to ensure that these blocks are in storage. The block loader can be in use for only one region at a time; to ensure this, IMS maintains a lock that is held by the region currently loading blocks from ACBLIB.

The workload wait sampler detected a BMP or JBP that needed the services of the block loader, but it was in use by another dependent region. This BMP or JBP region had to wait for the block loader, and, therefore, the transaction was suspended and the region was suspended.

The only way to reduce this contention for the block loader is to reduce the number of schedules performed by IMS or to reduce the time the block loader is used.

Scheduling activity may be reduced by:

- Using Wait-For-Input regions
- Increasing the PROCLIM value

The time the block loader is in use may be reduced by:

- Making some PSBs and DMBs RESIDENT
- Page fixing the PSBP and DMBP pools
- Increasing the size of the PSBP and DMBP pools

The use of the block loader and the contention for it is, to a large degree, intrinsic in the IMS design.

## SC: BLR Busy - DBT

Similar to “SC: BLR Busy - BMP” on page 484, but the scheduler is executing in a DBT region.

## SC: BLR Busy - MPP

The IMS MPP or JMP region scheduler was in the process of scheduling an MPP or JMP. Before the MPP or JMP can be called, the associated PSB and DMBs must be in the main storage pools. The IMS block loader is used to ensure that these blocks are in storage. The block loader can only be in use for one region at a time and to ensure this, IMS maintains a lock that is held by the region currently loading blocks from ACBLIB.

The workload wait sampler detected an MPP or JMP region that needed the services of the block loader, but it was in use by another dependent region. This MPP or JMP had to wait for the block loader and, therefore, the transaction was suspended and the MPP or JMP was suspended.

The only way to reduce this contention for the block loader is to reduce the number of schedules performed by IMS or by reducing the time the block loader is used. Scheduling activity may be reduced by:

- Using Wait-For-Input regions.
- Increasing the PROCLIM value. The time the block loader is in use may be reduced by:
- Making some PSBs and DMBs RESIDENT.
- Page fixing the PSBP and DMBP pools.
- Increasing the size of the PSBP and DMBP pools.

The use of the block loader and the contention for it is, to a large degree, intrinsic in the IMS design.

## SC: CBTS Latch

During the sampling of the system, IMS was scheduling a region and required the dynamic control block latch but the latch was in use by another IMS task.

## SC: CTL SYNC I/O Active

During a sampling of the system, the IMS scheduler, executing in the control region, had issued a request to OS/390 for an unidentified I/O operation. IMS issues miscellaneous I/Os that result in implied waits within OS/390. This can seriously impact IMS performance, because implied waits suspend the entire control task of IMS while OS/390 executes the I/O, such as directory searches (BLDL) of the ACBLIB, to locate a PSB:11.ACBLIB specified as DOPT.

## SC: CTL SYNC I/O Queued

The same as “SC: CTL SYNC I/O Active” above, except that the I/O cannot be started because of I/O subsystem contention. The I/O is queued on the logical channel or UCB.

## SC: DB Intent - BMP

During the initialization of a BMP or JBP region, IMS could not complete the setup, because the subject thread's PSB contained a PCB with PROCOPT=E, but the object database was already in use by another dependent region. Conversely, the BMP or JBP thread may not need exclusive use of the database, but a program is already scheduled with PROCOPT=E against the database. As a result, the region remains suspended until the database is no longer in use by other regions or until the owning PROCOPT=E program completes. The associated transaction is also suspended.

PROCOPT=E should be used judiciously and a high occurrence of this event indicates that it has been used inappropriately. A high occurrence of PI waits in the application programs indicates that PROCOPT=E should be considered.

## SC: DB Intent - DBT

Similar to "SC: DB Intent - BMP", but the scheduler is executing in a DBT region.

## SC: DB Intent - MPP

During the scheduling of an MPP or JMP, IMS could not complete the setup because the subject MPP's or JMP's PSB contained a PCB with PROCOPT=E, but the object database was already in use by another dependent region. Conversely, the MPP or JMP may not need exclusive use of the database, but a program is already scheduled with PROCOPT=E against the database.

As a result, the region remains suspended until the database is no longer in use by other regions or until the owning PROCOPT=E program completes. The associated transaction is also suspended.

PROCOPT=E should be used judiciously; a high occurrence of this event indicates that it has been used inappropriately. A high occurrence of PI waits in the application programs indicates that PROCOPT=E should be considered.

The SCHD parameter on the TRANSACT macro of the IMS SYSGEN tells IMS what to do if an intent failure occurs. Use of this parameter can prevent the region from remaining suspended if other work is in the input queue and can be processed in this region. The transaction that failed because of INTENT is placed back in the queue and another message is selected for scheduling.

## SC: DBWP Pool - BMP

Once an application program starts executing, it issues DL/I calls. DL/I requires a work area for accessing the databases. The DB work pool is used for this purpose. The ACBGEN process placed the DB work pool requirement for a given DMB in the DMB itself.

As part of the scheduling process, the block loader reserves an area in the DB work pool of size equal to the sum of all the DMBs the PSB is sensitive to. If there is insufficient room in the pool, the scheduler waits for space.

As a general rule, waits of this type should never occur if the DB work pool has been sized accurately. IBM's ADF product and Data Dictionary product can require substantial space in this pool.

## **SC: DBWP Pool - DBT**

Similar to “SC: DBWP Pool - BMP”, but the scheduler is executing in a DBT region.

## **SC: DBWP Pool - MPP**

See “SC: DBWP Pool - BMP”.

## **SC: DC Mon Latch**

During a sampling of the system, IMS was processing a message and needed the DC Monitor latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The DC monitor latch serializes the use of the DC Monitor in IMS. Whenever the DC Monitor is active, each call for it to move a new record to the buffers requires this latch. A high occurrence of this event indicates that the DC Monitor is very busy or that the DC Monitor is hung up in its I/O attempt to the DASD or tape file.

## **SC: DMB I/O Active**

During a sampling of the system, the IMS scheduler, executing in the control region, had issued a request to OS/390 to read in a DMB for a database needed by the program being scheduled. The ACBGEN utility builds a DMB from the DBD in DBDLIB. The DMB describes the structure of a database.

Some ways to reduce DMB I/O requirements are:

- Make the DMB RESIDENT using the DATABASE macro
- Reduce the number of different DMBs by creating common ones
- Increase the size of the DMBP pool to allow more DMBs to be in the pool concurrently

## **SC: DMB I/O Queued**

The same as “SC: DMB I/O Active”, except that the I/O to load the DMB from ACBLIB cannot be started because of I/O subsystem contention. The I/O is queued on the logical channel or UCB.

## SC: DMBP Pool - BMP

Before a transaction can be processed, all the databases specified in the associated PSB must have their DMBs in the DMB pool. During the scheduling process, the block loader ensures that all DMBs are in main storage.

During a sampling, a transaction was found that was in the process of being scheduled, but at least one DMB was not already in main storage and there was insufficient room in the pool to bring it in. The scheduling process for this transaction is suspended until there is enough room in the pool.

There are several ways to reduce DMB pool failures:

- Make some of the DMBs RESIDENT (specify RESIDENT on the DATABASE macro and RES=Y on the IMS control region JCL). These DMBs are brought into virtual storage at control region startup in an area separate from the DMB pool.
- Increase the size of the pool so that most DMBs in concurrent use can fit in the pool.

DMBs are reentrant control blocks, and, therefore, all regions accessing the same database use the same DMB. No more than one copy of a DMB is ever in storage at one time.

A high price is paid if the block loader deletes unused DMBs from the pool to make room for a new one. First, the DMB being freed requires that the database be CLOSED because the database's DCB is contained within the DMB. Second, the new DMB being brought into the pool requires that the associated database be OPENed.

## SC: DMBP Pool - DBT

Similar to “SC: DMBP Pool - BMP”, but the scheduler is executing in a DBT region.

## SC: DMBP Pool - MPP

See “SC: DMBP Pool - BMP”.

## SC: FETCH I/O Active

During a sampling of the system, the IMS scheduler, executing in the dependent region, had issued a request to OS/390 to load the application program. OS/390 program fetch had issued I/O to STEPLIB or LNKLIB, and the I/O is active.

The application program must be fetched for each schedule unless it is preloaded or brought in through a virtual fetch. Program fetch often accounts for a significant part of a transaction time-in-system. Some ways to reduce fetch I/O requirements are:

- Preload highly used programs
- Use wait-for-input programs
- Use virtual fetch
- Run highly used MPPs in BMP mode (or JMPs in JBP mode).

## SC: FETCH I/O Queued

The same as “SC: FETCH I/O Active” on page 488, except that the program fetch I/O cannot be started because of I/O subsystem contention. The I/O is queued on the logical channel or UCB.

## SC: Generic Latch

During a sampling of the system, IMS was processing a message and needed a generic latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

A generic latch is one that locks a caller-defined entity. Any part of IMS that requires serialization but is not defined under one of the other specific latches is locked by a generic latch. Generic latches are used to lock updates for the items in the following table:

Table 8. Latch Locks

Generic Latch	Latch Target
DMBE	Data management block (DMB) updates
VSBP	VSAM buffer pool master updates
DBBP	OSAM buffer pool master updates

## SC: IMS-CTL CPU Wait

The same as “SC: IMS-Dep CPU Wait” on page 490, except that the scheduler is waiting for a CPU in the control region.

## SC: IMS-CTL CSA Fault

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in CSA while executing under the control task in the control region.

## SC: IMS-CTL CSA GFA

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in CSA while executing under the control task in the control region.

## SC: IMS-CTL LPA Fault

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in LPA while executing under the control task in the control region.

## SC: IMS-CTL LPA GFA

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in LPA while executing under the control task in the control region.

## **SC: IMS-CTL Prv Fault**

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in the private area of the control region while executing under the control task in the control region.

## **SC: IMS-CTL Prv GFA**

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in the private area of the control region while executing under the control task in the control region.

## **SC: IMS-CTL Using CPU**

The same as “SC: IMS-Dep Using CPU” on page 491, except that the scheduler is active on a CPU in the control region.

## **SC: IMS-Dep CPU Wait**

The scheduler was executing in the dependent region and was not waiting for any event to occur. The scheduler work could be executed, but there were no CPUs available to process it. In the case of a UP processor, the CPU was executing other work (IMS or non-IMS) in the system. For MP processors, all the CPUs were busy with other work.

From an OS viewpoint, the dependent region program control task was on the OS dispatcher work-to-do queue. A high occurrence of this event indicates that the CPU is not fast enough or that too much higher-priority work was using the CPU cycles.

## **SC: IMS-Dep CSA Fault CSA**

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in CSA.

## **SC: IMS-Dep CSA GFA CSA**

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in CSA.

## **SC: IMS-Dep LPA Fault**

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in LPA.

## **SC: IMS-Dep LPA GFA**

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in LPA.

## **SC: IMS-Dep Prv Fault**

During a sampling of the system, the IMS scheduler was processing a message and suffered a page fault. Processing of the message is delayed until the fault is resolved.

For the scheduler, this indicates that a particular IMS scheduler task (ITASK) in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

## **SC: IMS-Dep Prv GFA**

During a sampling of the system, the IMS scheduler was processing a message and suffered a page fault. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed because no real frames were available. The page fault is placed on the general frame allocation (GFA) queue while the situation is dealt with.

For the scheduler, this indicates that a particular IMS scheduler task (ITASK) in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## **SC: IMS-Dep Using CPU**

The scheduler was executing in the dependent region and was not waiting for any event to occur. The scheduler work could be executed and, in fact, a CPU was actively processing it. In the case of a UP processor, the CPU was executing this scheduler task. For MP processors, one of the CPUs was executing the scheduling task.

A high occurrence of this event indicates that the scheduling parameters in the IMS SYSGEN are so involved that the scheduler spends a lot of CPU cycles to prepare a region for execution.

## **SC: IMS-SAS CPU Wait**

The same as “SC: IMS-Dep CPU Wait” on page 490, except that the scheduler is waiting for a CPU in the DL/I subordinate address space under the DL/I serial task.

## **SC: IMS-SAS CSA Fault**

The same as “SC: IMS-Dep Prv Fault”, except that the scheduler was accessing data or instructions in CSA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SC: IMS-SAS CSA GFA**

The same as “SC: IMS-Dep Prv GFA”, except that the scheduler was accessing data or instructions in CSA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SC: IMS-SAS LPA Fault**

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in LPA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SC: IMS-SAS LPA GFA**

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in LPA while executing in the DL/I subordinate address space.

## **SC: IMS-SAS Prv Fault**

The same as “SC: IMS-Dep Prv Fault” on page 491, except that the scheduler was accessing data or instructions in the private area of the DL/I subordinate address space while executing under the DL/I serial task in the DL/I subordinate address space.

## **SC: IMS-SAS Using CPU**

The same as “SC: IMS-Dep Using CPU” on page 491, except that the scheduler is active on a CPU in the DL/I subordinate address space under the DL/I serial task.

## **SC: IMS-SAS Prv GFA**

The same as “SC: IMS-Dep Prv GFA” on page 491, except that the scheduler was accessing data or instructions in the private area of the DL/I subordinate address space while executing under the DL/I serial task in the DL/I subordinate address space.

## **SC: Int List I/O Active**

During a sampling of the system, the IMS scheduler, executing in the control region, had issued a request to OS/390 to read in the intent list for the program to be scheduled. The ACBGEN utility builds an intent list for each PSB. The intent list is used by IMS to determine if the application program's intent against its databases is in conflict with any other currently active PSBs.

Some ways to reduce intent list I/O requirements are:

- Make the intent lists RESIDENT using the APPLCTN macro
- Reduce the number of different PSBs by creating common ones
- Increase the size of the PSBP pool to allow more intent lists to be in the pool concurrently

## **SC: Int List I/O Queued**

The same as “SC: Int List I/O Active”, except that the intent list I/O cannot be started because of I/O subsystem contention. The I/O is queued on the logical channel or UCB.

## **SC: IRLM Wait**

During a sampling of the system, the IMS scheduler, executing in the control region, had issued a request to the IRLM because of database block level sharing. This request resulted in a wait.

## SC: LGMSG I/O Active

The same as “IC: LGMSG I/O Active” on page 469, except that the LGMSG I/O is because of a request from the IMS scheduler.

## SC: LGMSG I/O Queued

The same as “IC: LGMSG I/O Queued” on page 470, except that the LGMSG I/O is because of a request from the IMS scheduler.

## SC: Log Buffer Wait

During a sampling, it was found that a buffer was not available for the current record to be logged and the scheduling task that created the record was waiting until a buffer was written to the data set.

Every major event that occurs in IMS results in a record being written to the IMS log. From this log, IMS supports recovery and accounting. This log is usually on tape to handle the volume produced.

Because of the critical nature of this data, the logging routines cause any task requesting their services (write a record describing an event) to wait until the logging can be completed. Records are posted to an in-storage buffer, which is usually written to the log only when full. While one buffer is being written, posting can continue to another buffer, if available.

If many log buffer waits are occurring, the buffer size and/or number of buffers should be increased. Contention in the logging function can have a critical performance impact.

## SC: Logical Log Latch

During a sampling of the system, IMS was processing a message and needed the logical logger latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The logical logger latch serializes the use of the logical logger in IMS. The logical logger is that part of IMS responsible for maintaining the log buffers before they are written to IEFRDER and IEFRDER2. Originally, this latch was used for only this purpose; however, in recent releases, this latch is used any time part of IMS wants to serialize the entire online system. Consequently, a high occurrence of logical latch waits is usually a sign that something is wrong outside of IMS. For example, if the log tape drive drops ready, the first sign of this in IMS is that the log buffers are all full and each task, as it attempts to place its log record in a log buffer, must wait for this latch.

## SC: LWA Wait

During a sampling of the system, schedule processing was in a LOG WRITE AHEAD (LWA) wait status. It was waiting for a log record to be physically written to the log before processing could continue. This is done to ensure data integrity in case of failure.

If necessary, IMS will write the log buffer containing this record, even if the buffer is not yet full, which is the normal signal for the I/O operation (see “SC: Log Buffer Wait” for more information about logging).

LOG WRITE AHEAD calls generally occur at sync point, which is when most physical database updates occur. They may also occur during application processing if IMS needs to use a database buffer that contains changed database records to access a different database block (buffer steal). The number of buffer steals may be reduced by increasing the number of buffers in these database subpools.

The DCLWA keyword in the IMS SYSGEN is used to activate or deactivate DC LWA.

### **SC: Misc IMS-CTL Wait**

Similar to “SC: Misc IMS-Dep Wait”, but the scheduler is executing in the control region.

### **SC: Misc IMS-Dep Wait**

Part of the scheduling process is executed in the control region and part is executed in the dependent region.

This event is detected when the sampler finds the IMS scheduler is scheduling a region and the scheduler was executing in the dependent region, but was waiting in the IMS dispatcher for an event to complete. The sampler could not attribute the wait to any other category in the scheduler component. An IWAIT was issued by the scheduler while executing in the dependent region.

### **SC: Misc IMS-SAS Wait**

Similar to “SC: Misc IMS-Dep Wait”, but the scheduler is executing in the DL/I subordinate address space (DLISAS) under the DL/I serial task.

### **SC: PSBP Pool - BMP**

Before a transaction can be processed, the program's PSB must be in the PSB pool. As part of the scheduling process, the PSB is brought into the pool if it is not already there.

During a sampling, a transaction was found that was in the process of being scheduled, but the PSB was not already in main storage and there was insufficient room in the pool to bring it in. The scheduling process for this transaction is suspended until there is enough room in the pool.

There are several ways to reduce PSB pool failures:

- Make some of the PSBs RESIDENT (specify RESIDENT on the APPLCTN macro and RES=Y on the IMS control region JCL). These PSBs are brought into virtual storage at control region startup in an area separate from the PSB pool.
- Increase the size of the pool so that most PSBs in concurrent use can fit in the pool.
- Reduce the number of schedules performed.
- Make judicious use of the PARALLEL scheduling option.

It should be noted that PSBs are non-reentrant control blocks and, therefore, if several regions are using the same PSB name, they all require their own copy of the PSB, each one using space in the PSB pool. In addition, if a PSB is defined as RESIDENT and PARALLEL, the resident copy will never be used. Instead, a copy of it will be made in the PSB pool for each region using it. This must be considered when calculating PSB pool size.

## **SC: PSBP Pool - DBT**

Similar to “SC: PSBP Pool - BMP” on page 494, but the scheduler is executing in a DBT region.

## **SC: PSBP Pool - MPP**

See “SC: PSBP Pool - BMP” on page 494.

## **SC: PSBW Pool - BMP**

Once an application program starts executing, it issues DL/I calls. DL/I requires a work area for handling these calls. The PSB work pool is used for this purpose. The ACBGEN process placed the PSB work pool requirement for a given PSB in the PSB itself.

As part of the scheduling process, the block loader reserves an area in the PSB work pool of the size found in the PSB. If there is insufficient room in the pool, the scheduler waits for space.

As a general rule, waits of this type should never occur if the PSB work pool has been sized accurately. IBM's ADF product and Data Dictionary product can require substantial space in this pool.

## **SC: PSBW Pool - DBT**

Similar to “SC: PSBW Pool - BMP”, but the scheduler is executing in a DBT region.

## **SC: PSBW Pool - MPP**

See “SC: PSBW Pool - BMP”.

## **SC: OS Wait-CTL**

Similar to “SC: OS Wait-Dep”, but the scheduler is executing in the control region.

## **SC: OS Wait-Dep**

Part of the scheduling process is executed in the control region and part is executed in the dependent region.

This event is detected when the sampler finds the IMS scheduler is scheduling a region and the scheduler was executing in the dependent region, but was waiting for the completion of an OS event. The sampler was unable to attribute the wait to any other category in the scheduler component. A wait was issued by the scheduler while executing in the dependent region. The wait may have been issued by a component of OS called by the IMS scheduler.

## **SC: OS Wait-SAS**

Similar to “SC: OS Wait-Dep”, but the scheduler is executing in the DL/I subordinate address space under the DL/I serial task.

## SC: Other Latch

During a sampling of the system, IMS was processing a message and needed a specific IMS latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The data collector could not determine which latch was needed. Call BMC Software Customer Support if this event occurs frequently.

## SC: PI ENQ/DEQ Latch

During a sampling of the system, a program being scheduled in a dependent region required the enqueueing of a database segment. The latch used to serialize the IMS routines that enqueue segments was already in use; this region had to wait.

During the processing of calls against databases, IMS ensures that database integrity is maintained by enqueueing database segments to prevent multiple updating. This is called program isolation (PI) and is one of the mechanisms used by IMS to achieve its high level of data integrity.

This event is not the same as a PI wait. A PI wait occurs when an application program requests access to data that is already in use by another program. A PI latch wait occurs when an application program requests access to data and DL/I must check the enqueue status of the data but the enqueue routine is already in use by another program.

A high occurrence of this event indicates that IMS is processing a very large number of PI enqueues and dequeues, or many page faults are occurring while it is accessing the PI pool.

## SC: PSB I/O Active

During a sampling of the system, the IMS scheduler, executing in the control region, had issued a request to OS/390 to read in the PSB for the program to be scheduled. The ACBGEN utility builds a PSB from the PSB in PSBLIB. The PSB describes the application program's access parameters for all the databases that the program is sensitive to. The PCBs are contained in the PSB.

Some ways to reduce PSB I/O requirements are:

- Make the PSB RESIDENT using the APPLCTN macro
- Reduce the number of different PSBs by creating common ones
- Increase the size of the PSBP pool to allow more PSBs to be in the pool concurrently

## SC: PSB I/O Queued

The same as "SC: PSB I/O Active", except that the I/O to load the PSB from ACBLIB cannot be started because of I/O subsystem contention. The I/O is queued on the logical channel or UCB.

## SC: QBLKS I/O Active

The same as "IC: QBLKS I/O Active" on page 474, except that the QBLKS I/O is because of a request from the IMS scheduler.

## **SC: QBLKS I/O Queued**

The same as “IC: QBLKS I/O Queued” on page 474, except that the QBLKS I/O is because of a request from the IMS scheduler.

## **SC: SAS SYNC I/O Active**

During a sampling of the system, the IMS scheduler, executing under the DL/I serial task in the DL/I subordinate address space, issued a request to OS/390 for an unidentified I/O operation. IMS issues miscellaneous I/Os that result in implied waits within OS/390. This can seriously impact IMS performance because implied waits hang the DL/I serial task of IMS while OS/390 executes the I/O, such as directory searches (BLDL) of the ACBLIB, to locate a PSB specified as DOPT.

## **SC: SAS SYNC I/O Queued**

The same as “SC: SAS SYNC I/O Active”, except that the I/O cannot be started because of I/O subsystem contention. The I/O is queued on the logical channel or UCB.

## **SC: SHMSG I/O Active**

The same as “IC: SHMSG I/O Active” on page 475, except that the SHMSG I/O is because of a request from the IMS scheduler.

## **SC: SHMSG I/O Queued**

The same as “IC: SHMSG I/O Queued” on page 476, except that the SHMSG I/O is because of a request from the IMS scheduler.

## **SC: Storage Mgt Latch**

During a sampling of the system, IMS was processing a message and needed the storage management latch but the latch was already in use by another IMS task. Processing of the message is suspended until the latch becomes available.

The storage management latch is used by the main storage management routines in IMS to create and destroy buffer pools and to get and free buffers within those pools. Because of the design of this part of IMS, storage management must be serialized; thus, this latch.

A high occurrence of this event is a sign that IMS is in extreme trouble because storage management executes quite quickly for each request and usually does not hang. Any requests for storage that require a wait (IWAIT) free the latch before waiting; therefore, pool space failures do not result in a high occurrence of this latch event.

---

## Synchronization Point (SP) Events

Synchronization point (sync point) processing is the processing done by IMS to maintain integrity in the system for a given transaction. It is instigated by the application program through a GU to the I/O PCB, by the application program through a CHKP or SYNC call, or when an application program ends. Sync point processing completes when all the database changes have been physically written to the databases and all message traffic has been appropriately discarded or made available for transmission (the input message is discarded and the output messages are ready to send).

IMS can execute sync processing twice for one transaction. This can occur when the application program completes the processing of the first message, then issues a GU to the I/O PCB to retrieve the second message, which results in a QC status code. At this point, the application normally would terminate. Two sync points will have occurred: one when the GU with the QC was processed; one when the application ended. The workload wait sampler will take this properly into account when sampling the system.

The following subsections list and define the events that may be shown for SYNC POINT by the DWAIT display. The descriptions are in alphabetical order.

### SP: CBTS Latch

During a sampling of the system, IMS was processing a sync point and required the dynamic control block latch but the latch was in use by another IMS task.

### SP: CTL SYNC I/O Active

The same as “SC: CTL SYNC I/O Active” on page 485, except that it occurred during sync point processing.

### SP: CTL SYNC I/O Queued

The same as “SC: CTL SYNC I/O Queued” on page 485, except that it occurred during sync point processing.

### SP: DB2S-DEP CPU Wait

The same as “AP: DB2N-Dep CPU Wait” on page 453 except that the DB2 interface code was processing a service call during IMS sync point processing.

A service call is issued by the IMS DB2 interface code for certain events in the dependent region. A CREATE THREAD call is issued to identify the transaction to DB2 and a COMMIT PHASE 1 and PHASE 2 call is done at IMS sync point time for data integrity.

**Note:** Only SQL calls made through the IMS attach (for example, BMPs or MPPs) are monitored. SQL calls made through the CICS attach are not shown.

### SP: DB2S-DEP CSA Fault

The same as “AP: DB2N-Dep CSA Fault” on page 453 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP CSA GFA**

The same as “AP: DB2N-Dep CSA GFA” on page 453 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP LPA Fault**

The same as “AP: DB2N-Dep LPA Fault” on page 453 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP LPA GFA**

The same as “AP: DB2N-Dep LPA GFA” on page 453 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP Prv Fault**

The same as “AP: DB2N-Dep Prv Fault” on page 453 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP Prv GFA**

The same as “AP: DB2N-Dep Prv GFA” on page 454 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP Using CPU**

The same as “AP: DB2N-Dep Using CPU” on page 454 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DB2S-DEP Wait**

The same as “AP: DB2N-Dep Wait” on page 454 except that the DB2 interface code was processing a service call during IMS sync point processing.

### **SP: DC Mon Latch**

The same as: “SC: DC Mon Latch” on page 487, except that it is the sync point manager that needs the latch.

### **SP: DEDB Area Latch**

During a sampling of the system, that one or more transactions were waiting for a specific DEDB area.

### **SP: DEDB Latch**

During a sampling of the system, one or more transactions were waiting for a DEDB latch.

## **SP: FP Syncpoint Latch**

During a sampling of the system, it was noted that one or more transactions were waiting for a Fast Path synchronization point latch.

## **SP: Generic Latch**

The same as “SC: Generic Latch” on page 489, except that it is the sync point manager that needs the latch.

## **SP: IMS-CTL CPU Wait**

The same as “SP: IMS-Dep CPU Wait” on page 501, except that the IMS sync point manager is waiting for a CPU in the control region.

## **SP: IMS-CTL CSA Fault**

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in CSA while executing under the control task in the control region.

## **SP: IMS-CTL CSA GFA**

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in CSA while executing under the control task in the control region.

## **SP: IMS-CTL LPA Fault**

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in LPA while executing under the control task in the control region.

## **SP: IMS-CTL LPA GFA**

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in LPA while executing under the control task in the control region.

## **SP: IMS-CTL Prv Fault**

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in the private area of the control region while executing under the control task in the control region.

## **SP: IMS-CTL Prv GFA**

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in the private area of the control region while executing under the control task in the control region.

## SP: IMS-CTL Using CPU

The same as “SP: IMS-Dep Using CPU” on page 502, except that the IMS sync point manager is active on a CPU in the control region.

A high occurrence of this event indicates:

- Queue data set problems - The output messages generated by the application program are permanently committed to the queues during sync point in the control region. If there are insufficient queue buffers or excessive I/Os are being executed against the three queue data sets, high CPU utilization in the sync point manager while in the control region will be noted.
- Excessive sync points - The path length through IMS sync point is often nontrivial. If many sync points per minute are being performed, then sync point can become a critical point in the performance of a transaction. The number of sync points taken can be reduced by using `MODE=MULT` for some transactions, or by reducing the number of `CHKP` or `SYNC` calls issued by BMPs or JBPs.

## SP: IMS-Dep CPU Wait

The IMS sync point manager was executing in the dependent region and was not waiting for any event to occur. The sync point manager work could be executed but there were no CPUs available to process it. In the case of a UP processor, the CPU was executing other work (IMS or non-IMS) in the system. For MP processors, all the CPUs were busy with other work.

From an OS viewpoint, the dependent region program control task was in the OS dispatcher work-to-do queue. A high occurrence of this event indicates that the CPU is not fast enough or that excessive work of a higher priority was using the CPU cycles.

## SP: IMS-Dep CSA Fault

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in CSA.

## SP: IMS-Dep CSA GFA

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in CSA.

## SP: IMS-Dep LPA Fault

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in LPA.

## SP: IMS-Dep LPA GFA

The same as “SP: IMS-Dep Prv GFA”, except that the sync point manager was accessing data or instructions in LPA.

## **SP: IMS-Dep Prv Fault**

During a sampling of the system, the sync point manager was processing a message and suffered a page fault. Processing of the message is delayed until the fault is resolved.

The sync point manager executing in the dependent region address space under the program control task was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

## **SP: IMS-Dep Prv GFA**

During a sampling of the system, the sync point manager was processing a message and a page fault occurred. Processing of the message is delayed until the fault is resolved. The page-in I/O process could not be executed because no real frames were available. The page fault is placed in the general frame allocation (GFA) queue until the situation is resolved.

The sync point manager executing in the dependent region was accessing data or instructions in the private area of the dependent region address space when the fault occurred.

General frame allocation (GFA) page faults refer to the situation when a page fault occurs and the OS/390 real storage manager (RSM) did not have any available frames to allocate to the page fault. To find a frame to use, the system resources manager (SRM) is called to steal from existing users. This may result in a page-out operation that extends the page fault resolution time even more.

## **SP: IMS-Dep Using CPU**

The IMS sync point manager was executing in the dependent region and was not waiting for any event to occur. The sync point manager work could be executed and, in fact, a CPU was actively processing it. In the case of a UP processor, the CPU was executing this sync point task. For MP processors, one of the CPUs was executing the sync point task.

A high occurrence of this event indicates that many database buffers are being updated by the application program. The only sync point processing performed in the dependent region is the flushing of modified database buffers. If an excessive number of buffers are not being updated, VSAM, OSAM, or IOS may be involved in I/O error recovery, which would account for the high CPU utilization here.

## **SP: IMS-SAS CPU Wait**

The same as “SP: IMS-Dep CPU Wait” on page 501, except that the IMS sync point manager is waiting for a CPU under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS CSA Fault**

The same as “SP: IMS-Dep Prv Fault”, except that the sync point manager was accessing data or instructions in CSA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS CSA GFA**

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in CSA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS LPA Fault**

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in LPA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS LPA GFA**

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in LPA while executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS Prv Fault**

The same as “SP: IMS-Dep Prv Fault” on page 502, except that the sync point manager was accessing data or instructions in the private area of the DL/I subordinate address space while executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS Prv GFA**

The same as “SP: IMS-Dep Prv GFA” on page 502, except that the sync point manager was accessing data or instructions in the private area of the DL/I subordinate address space while executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: IMS-SAS Using CPU**

The same as “SP: IMS-Dep Using CPU” on page 502, except that the IMS sync point manager is active on a CPU under the DL/I serial task in the DL/I subordinate address space.

## **SP: IRLM Wait**

The same as “AP: IRLM Wait” on page 461, except that it occurred during sync point processing.

## **SP: LGMSG I/O Active**

The same as “IC: LGMSG I/O Active” on page 469, except that the sync point manager has initiated the request that caused the I/O.

## **SP: LGMSG I/O Queued**

The same as “IC: LGMSG I/O Queued” on page 470, except that the sync point manager has initiated the request that caused the I/O.

## **SP: Log Buffer Wait**

The same as “SC: Log Buffer Wait” on page 493, except that it occurred during sync point processing.

## **SP: Logical Log Latch**

The same as “IC: Logical Log Latch” on page 470, except that it is the sync point manager that needs the latch.

## **SP: LWA Wait**

The same as “AP: LWA Wait” on page 462, except that it occurred during sync point processing, not because of buffer steals.

Whenever the sync point manager is invoked on behalf of an application program (GU to I/O PCB, CHKP, SYNC, or program termination), the first activity performed is the writing of all database blocks or CIs modified by the application. This activity is performed in the dependent region to increase the parallelism of the sync point process across multiple regions. If LWA is active, the writing of the blocks is deferred until the associated log records (type 50) are physically on the log; therefore, the log is written ahead of the database blocks. LWA improves the recoverability of an IMS database, but reduces the performance of the transactions processed.

A high occurrence of this event indicates that the workload volume is insufficient to cause the log blocks to be written before a transaction reaches sync point. The negative impact of LWA can be reduced by reducing the block size of the log, because the smaller log blocks fill more quickly. A commensurate increase in log buffers should also be made to prevent delay of other transactions that need log buffer space. Often reducing the log block size to the absolute minimum allowed by IMS still results in high LWA waits in sync point. Little can be done by the IMS user to improve the situation.

## **SP: Misc IMS-CTL Wait**

The same as “SP: Misc IMS-Dep Wait”, except that the sync point manager is executing in the control region.

## **SP: Misc IMS-Dep Wait**

Part of the sync point process is executed in the control region and part is executed in the dependent region.

This event is detected when the sampler finds a transaction within sync point processing, the processing was occurring in the dependent region but was waiting in the IMS dispatcher for an event to complete. The sampler could not attribute the wait to any other category in the sync point component. An unknown IWAIT was issued by the IMS sync point manager while executing in the dependent region.

If this event is ever a significant contributor to response time, call BMC Software Customer Support.

## **SP: Misc IMS-SAS Wait**

The same as “SP: Misc IMS-Dep Wait”, except that the sync point manager is executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: MSDB Latch**

During a sampling of the system, one or more transactions were waiting for an MSDB latch.

## **SP: OS Wait-CTL**

The same as “SP: OS Wait-Dep”, except that the sync point manager is executing in the control region.

## **SP: OS Wait-Dep**

Part of the sync point process is executed in the control region and part is executed in the dependent region.

This event is detected when the sampler finds a transaction within sync point processing. The processing was occurring in the dependent region but was waiting for the completion of an OS event. The sampler could not attribute the wait to any other category in the sync point component. The WAIT may have been issued by a component of OS called by the IMS sync point manager.

The WAIT usually is caused by a BMP, JBP, MPP, or JMP region that issues its own STIMER WAIT macros. This technique was commonly used before WFI (wait-for-input) transactions became available. Another possibility is a transaction that issues a WTOR and waits for the reply. You may use the PGMTYPE or TRAN parameter to eliminate these programs or transactions from sampling.

## **SP: OS Wait-SAS**

The same as “SP: OS Wait-Dep”, except that the sync point manager is executing under the DL/I serial task in the DL/I subordinate address space.

## **SP: OSAM DB I/O Active**

During a sampling of the system, the sync point manager had issued a purge call to write out modified OSAM blocks. At the time of the sample, the I/O was active against the database. Processing of the message is suspended until the I/O completes.

## **SP: OSAM DB I/O Queued**

The same as “SP: VSAM DB I/O Active” on page 506, except that the OSAM database I/O was queued on the logical channel or UCB because I/O contention prevented the successful initiation of the SIO instruction.

## **SP: Other Latch**

The same as “SC: Other Latch” on page 496, except that it is the sync point manager that needs the latch.

## **SP: PI ENQ/DEQ Latch**

The same as “AP: PI ENQ/DEQ Latch” on page 465, except that the PI routine is called as part of sync point processing. Generally, the calls to the PI routines here are to release database enqueues acquired during the preceding application program processing.

## **SP: QBLKS I/O Active**

The same as “IC: QBLKS I/O Active” on page 474, except that the sync point manager has initiated the request that caused the I/O.

## **SP: QBLKS I/O Queued**

The same as “IC: QBLKS I/O Queued” on page 474, except that the sync point manager has initiated the request that caused the I/O.

## **SP: SAS SYNC I/O Active**

The same as “SC: SAS SYNC I/O Active” on page 497, except that it occurred during sync point processing.

## **SP: SAS SYNC I/O Queued**

The same as “SC: SAS SYNC I/O Queued” on page 497, except that it occurred during sync point processing.

## **SP: SHMSG I/O Active**

The same as “IC: SHMSG I/O Active” on page 475, except that the sync point manager has initiated the request that caused the I/O.

## **SP: SHMSG I/O Queued**

The same as “IC: SHMSG I/O Queued” on page 476, except that the sync point manager has initiated the request that caused the I/O.

## **SP: Storage Mgt Latch**

The same as “IC: Storage Mgt Latch” on page 476, except that it is the sync point manager that needs the latch.

## **SP: VSAM DB I/O Active**

During a sampling of the system, the sync point manager had issued a purge call to write out modified VSAM CIs. At the time of the sample, the I/O was active against the database. Processing of the message is suspended until the I/O completes.

## **SP: VSAM DB I/O Queued**

The same as “SP: VSAM DB I/O Active”, except that the VSAM database I/O was queued on the logical channel or UCB because I/O contention prevented the successful initiation of the SIO instruction.



---

## Part 4. Appendixes

**Appendix A. IMS Dump Analysis** .....511

AO Exit.....511

MAINVIEW AutoOPERATOR for IMS Routines in IMS .....511

Event Collector.....511

**Appendix B. How Product Libraries Should Be Used** .....513



---

## Appendix A. IMS Dump Analysis

This appendix describes how to analyze an IMS dump with MAINVIEW for IMS or MAINVIEW AutoOPERATOR for IMS installed.

---

### AO Exit

Except for MAINVIEW AutoOPERATOR for IMS MTO message capture for the Journal log, all AO code merely passes control to the specified routines during operation. During initialization, special protection exists while the Event Collector is being set up.

---

### MAINVIEW AutoOPERATOR for IMS Routines in IMS

During initialization, the MAINVIEW AutoOPERATOR for IMS AO code creates two subtasks under the IMS control task. Each of these is protected by ESTAE routines and uses different control blocks than IMS. These subtasks generally can be ignored during IMS dump analysis since they do not affect the IMS flow. These subtasks are terminated correctly at IMS termination.

---

### Event Collector

The following should be noted about IMS dumps:

- MAINVIEW for IMS register save areas are in MAINVIEW for IMS data areas, not in the IMS prechained save areas. The IMS chains remain unchanged.
- Sometimes the R14 return register in an IMS save area does not point back into the calling IMS module. The IMS R14 value can be found 4 bytes in front of the address pointed to by R14.
- A MAINVIEW for IMS module at entry saves the registers of an IMS module in the next IMS prechained save area pointed to by R13.
- MAINVIEW for IMS module registers are always saved in MAINVIEW for IMS save areas. These save areas are assigned dynamically as required. Normally, one of several preallocated save areas per region is used.

Each active IMS region has a MAINVIEW for IMS data area acquired for it at region initialization. This data area is in ECSA and is named `IMERDnnn`, where *nnn* is the PST number. The preallocated save areas are in this block.

If more save areas are required, a dynamic storage pool is used. This pool is also in ECSA and is named `IMFSP000`.

Any area in actual use as a MAINVIEW for IMS save area, whether in `IMERDnnn` or `IMFSP000`, is identified with SAR or ISA.

- In most cases, only a save area backward pointer exists, pointing from the MAINVIEW for IMS area to the previous IMS save area. While a MAINVIEW for IMS module is in control, the current R12 is its base register and the current R13 points to its SAR.

- In some cases, a MAINVIEW for IMS module transfers control to an IMS module, but needs to regain control after it completes processing. In this case also, the IMS save area chains remain unchanged.

The only sign of the presence of a MAINVIEW for IMS module is an R14 value in an IMS save area which does not point back to the calling IMS module. If you need to verify the IMS path, the IMS R14 value is saved in the MAINVIEW for IMS SAR, 4 bytes in front of the address pointed to by the R14 in the IMS save area.

**Note:** When MAINVIEW for IMS interfaces between two IMS modules, MAINVIEW for IMS is transparent to the IMS modules. All registers are preserved.

- IMECSRvx and IMFCSRvx appear in dumps as active ITASKS. This is normal as long as the current save area is for DFSIWAIT.

---

## Appendix B. How Product Libraries Should Be Used

Several distributed libraries are included with your MAINVIEW products, including a parameter library (BBPARM), a sample library (BBSAMP), and a profile library (BBPROF). Use the contents of these distributed libraries as models to create site-customized product libraries, either manually or automatically, with AutoCustomization.

### Warning

The distributed libraries should never be modified. If you change the distributed libraries, subsequent SMP maintenance will overwrite your changes.

Throughout the MAINVIEW documentation set, references to these libraries use the distributed name. However, when you need to make changes, be sure to use the corresponding library that has been customized for your site. Table 9 lists the distributed name, the corresponding customized library created by AutoCustomization, and leaves space for you to note any other corresponding library that may have been created for your site.

Table 9. Product Libraries

Distributed Library Name	Library Created by AutoCustomization	Other Site-Customized Copy
BBPARM	UBBPARM	
BBSAMP	UBBSAMP	
BBPROF	SBBPROF	

For more detailed information about all the product libraries, see “Using MAINVIEW Product Libraries” in the *MAINVIEW Common Customization Guide* or “Using Product Libraries” in the *MAINVIEW Administration Guide*.



---

# Index

## Symbols

\$\$IMSDIR 85  
(TOTAL) 56, 57  
+ qualifier  
    ISTAT request 396  
    MWAIT request 441  
    USER request 427  
    workload analyzer request 75  
. 18  
> indicator  
    DREGN display 128, 148

## A

activity  
    call 127, 147  
    DB2 127  
    paging 148  
    program isolation 128, 148  
    system 128, 148  
ACTL latch 277  
address space 140, 160  
ALL (SET operand) 439  
allocation  
    buffer 225  
    failures 256  
    fetch request elements (FREs) 84  
    fixed pools  
        size percentage 264  
    normal buffer 136, 157  
    overflow buffer 136, 157  
    pools 295  
        size percentage 295  
    requests 235, 243  
        to APPC 256  
    variable pools  
        size percentage 269  
allocation directions  
    input 237, 243, 249  
    LU 246  
    output 235, 243, 254  
alters 230  
alters requests  
    OSAM subpool 220  
analyzer  
    locks  
        DREGN program isolation activity 139, 158  
        PI service 284  
    services 63  
AO exit 511  
AOIP pool  
    DPOOL display 265  
    POOLS display 296  
AP  
    DWAIT application program events 451

APPC  
    allocation requests 256  
    connection status 238, 245  
    EXPAND 237  
    LUMC pool 265, 297  
    LUMP pool 265, 297  
    mode name 252, 257  
    startup parameter 239, 246  
APPCA 244, 250, 255  
APPCL 237, 250, 255  
    EXPAND 244  
APPLCTN macro 11  
application program  
    DWAIT AP events 451  
APSB latch 274  
asynchronous conversations 238, 245  
asynchronous notify request 309, 315, 349, 356  
asynchronous write requests  
    logger 281  
authorization ID  
    DB2 128  
average data measurement 54  
AWE write request 281

## B

background write 227, 232  
backup request 309, 316, 350, 356  
basic conversation 251, 257  
BBIISP00  
    interval keyword 436  
BBI-SS PAS 52  
    Image log 61  
        analyzer displays 60  
        monitor displays 60  
    Journal log 55, 57  
BBPARM  
    changing 513  
BBPARM data set  
    time interval default 436  
BBPARM/UBBPARM considerations 513  
BBPROF/SBBPROF considerations 513  
BBSAMP/UBBSAMP considerations 513  
BID 322, 331, 344, 363, 373, 389  
BLKDBA2 35  
BLKIMFW 62  
BLKM latch 277  
BLKMGR 32  
B-lock 322, 331, 343, 363, 373, 389  
block  
    create requests  
        OSAM buffer handler 216  
        OSAM subpool 220  
    lock 322, 331, 344, 363, 373, 389

- block (continued)
  - reads
    - OLDS 281
  - size
    - OLDS 279
    - queue data sets 89
    - WADS 279
  - space
    - MFS 88
  - writes
    - OLDS 281
- blocking requests 62
- blocks
  - MFS 88
  - new OSAM 217
  - OSAM subpool
    - written 220
  - total OSAM written 217
- BMP
  - block loader busy 484
  - database intent 486
  - DBWP pool 486
  - GU 477
  - not active 477
  - unschedulable 477
- BTAM 474
  - MFS 472
  - save area prefix (SAP) 474, 475
- buffer
  - error 217, 231
    - VSAM 231
  - Fast Path
    - fixed availability 224
    - size 224
  - fixed
    - pools 267
  - handler
    - number of OSAM requests 216
    - waits 221
  - index 230
  - invalidation
    - IRLM notify 311, 317, 351, 357
    - PI 287
  - latch
    - overflow 465
  - log wait 470
  - OSAM
    - free space 222
    - performance 217
  - overflow 323, 332, 345, 364, 374, 390
    - Fast Path 286
  - page fixed 224, 225, 230
  - pool hit ratios
    - DBST OSAM global pool display 217
    - DBST OSAM subpool display 221
  - pool performance 228, 231
  - pool utilization 215

- buffer (continued)
  - pools
    - Fast Path 223
    - OSAM 215, 219
    - subpool size 228, 230
    - usage
      - DPOOL 292
      - WADS 281
  - buffer handler
    - work performed 217
  - buffer handler trace 216, 227

## C

- call activity 127, 147
- calls
  - DL/I
    - DAPPC 253
  - GET NEXT within PARENT 134
  - SQL
    - DAPPC 253
  - STAT 124
  - token 124
- CBSINTVL parameter 396
- CBSORTSZ parameter 396
- CBT pools 24
  - summary 288
- CBTS
  - latch 452, 485, 498
- CBTS latch 274
- CCTL
  - LATCH 275
- CESS pool
  - description 26
  - DPOOL display 265
  - POOLS display 296
- change request 309, 315, 349, 356
- channels
  - database I/O queuing 466, 507
- check write requests 281
- checkpoint
  - calls 227
  - cold start 283
  - log records 283
- CHKP 498
- CICS
  - application program 452
- CIOP pool
  - DPOOL display 265
  - POOLS display 296
  - tuning 27
- CLASQ
  - sample request 34
- class
  - queuing 8
  - SET keyword 437
- COLD (SET operand)
  - restart
    - MWAIT service 437

- cold start 247, 256
  - checkpoint 283
- collecting data on an application 434
- color
  - APPCA 233
  - APPCL 242
  - DAPPC 249, 254
  - USER 427
- COMCYCL 318
- COMMIT 454, 498
- communication
  - I/O buffer pool 265, 297
- configuration
  - data set 513
  - log data sets 279
- confirmations
  - TP 256
- connection
  - APPC
    - current status 238, 245
    - user 241, 252
- contention
  - IRLM request 348, 355, 359
  - latches 258
  - OSAM block segment 286
- control interval
  - VSAM 228
- control region
  - CPU wait 489
  - LPA fault 489
- control task
  - wait 466
- CONV latch 275
- conversation
  - active 246, 250, 255
    - users 428
  - asynchronous 238, 245
  - basic 251, 257
  - ID 235, 240
  - initiation
    - TP name 241, 251, 257
  - input 239, 246
  - level 251, 257
  - mapped 251, 257
  - output 239, 246
  - synchronous 239, 246
  - transaction 138, 157
  - type 251, 257
  - users
    - held 428
- count data measurement 54
- CPI-C
  - DAPPC (input allocation) 253
  - send requests 252, 257
- CPU
  - service units 140, 160
- CPU time
  - TCB 143, 162

- CPU wait 468
  - application program 451
  - control region 489
  - dependent region 490
  - SAS 491
- CREATE THREAD 454, 498
- cross-memory program call
  - IRLM 308, 314, 348, 354
- CSA
  - fault 468
    - application program 451
    - DB2 455
    - DL/I 457
    - sync point manager 502
- CURPER
  - default 433
  - SET keyword 436
  - use 441
- current period 436, 447
- cursor-sensitive fields 237
- customer support iii
- cylinders
  - formatted OSAM 217

## D

- DAPPC
  - EXPAND 250, 255
- DASD
  - I/O 469
- data capture exit 119
- data collection
  - monitor list 52
- data entry panel
  - workload wait request 435
- data sampling
  - MWAIT 441
- data set 89
  - configuration 513
  - extension 287
  - long message queue 90
  - QBLKS 90
  - queue 90
- data sharing
  - group 347, 354
- data sharing group 350, 357
- data type 54
- database
  - activity 215
  - analysis 16
  - intent 9
    - BMP 486
    - DBT 486
    - MPP 486
  - lock 307
  - sharing 371, 381, 388
- Database Administrator 35

- DB2
  - activity 127
  - application program (AP) events 453
  - application program wait 454
  - authorization ID 128
  - connection status
    - DAPPC 253
  - CPU time
    - DWAIT events 453, 454, 455
  - CPU wait 454
  - CSA fault 455
  - LPA fault 455
  - page fault 453
  - resource contention 454
  - SET operand 439
  - synchronization point (SP) events 498
- DBA (Database Administrator) 35
- DBAU latch 275
- DBBP
  - latch 276, 469
  - lock 461, 489
- DBCTL
  - data collection 433
  - LATCH 275
- DBFHAGU0 439
- DBHIT 17
- DBLK latch 275
- DBLR latch 274
- DBP
  - DBWP pool 487
- DBRC 287
- DBSL latch 275
- DBST 215
- DBST display 17
- DBSTL
  - sample request 33
- DBT
  - block loader busy 485
  - database intent 486
- DBT (SET operand) 438
- DBTOT
  - example 35
- DBWP pool
  - BMP 486
  - DBT 487
  - DPOOL display 268, 270
  - MPP 487
  - POOLS display 297
- DC monitor 469
  - latch 455, 487
- DCB
  - number 322, 331, 343, 363, 373, 388
- DCLWA 282
- DCSL latch 275
- DDRB latch 275
- DDRP latch 275
- dead letter
  - user queues 428
- deadlock 23
  - cycles 318, 358
  - exit 310, 317, 351, 357
  - global 318, 358
  - local 318, 358
  - suspends 312, 319
- deallocation requests 256
- DECB
  - wait 93
- DEDB 18
  - access control 275
  - area name 322, 331, 343, 363, 373, 388
  - latch 499
  - resource ID 286
  - synchronization control 275
- DEDB events
  - control interval contention 456
  - latch 456
  - resource latch 456
- delayed transactions 450
- dependent regions
  - CPU wait 490
- detail measurement 53
- detail region displays 125, 145
- device type
  - queue data set 90
- DFSCLM00 276
- DFSCLKX0 310, 317, 351, 357
- DFSHASH0
  - PI service 286
- DFSISMN0 24
- DFSLU 241, 247, 250, 255
- DFSNOTX0 311, 317, 351, 357
- DFSRESX0 311, 317, 351, 358
- directory
  - MFS
    - I/O active 472
    - I/O queued 472
  - reads 85
- DISP latch 275
- display service
  - ISTAT 395
  - logging 60, 61
  - TRANQ 423
  - Workload Wait 445
- DL/I
  - buffer handler trace
    - DBST 216
    - VSST 227
  - calls
    - DAPPC 253
    - details 119
    - DLIST 118
    - CPU wait 457
  - cross memory
    - CPU wait 460
    - CSA fault 460
    - LPA fault 460
  - DREGN 127, 147

- DL/I (continued)
  - LSO 458
  - PCB 123
  - SAS 459
  - trace table
    - storage 227
- DL/I activity 14–15
- DLDP pool
  - DPOOL display 268, 270
  - POOLS display 297
- DLI
  - SET operand 439
- DLISAS
  - data sets allocated 34
  - POOLS display
    - CBT pools 290
    - wait 494
- DLIST
  - use 14
- DLMP pool
  - DPOOL display 268, 270
  - POOLS display 297
- DLTCH
  - generic detail statistics 258
- DMB
  - I/O active 487
  - I/O queued 487
- DMB pool 26
  - intent failure 9
  - tuning 11
- DMBB latch 275
- DMBE
  - latch 469
  - lock 461, 489
- DMBP latch 275
- DMBP pool
  - BMP 488
  - DBT 488
  - DPOOL display 270
  - MPP 488
  - POOLS display 297
- DMBW pool
  - DPOOL display 270
  - POOLS display 297
  - use 26
- DMON
  - monitor service title 56
  - use 3
- DMSH latch 275
- DMSY latch 275
- DOPT option 11
- DPOOL 292
- DPSB pool
  - DPOOL display 268, 270
  - POOLS display 297
- DPST pool
  - tuning 24
- DREGN 250
  - description
    - Event Collector data 125
    - No Event Collector data 145
  - program isolation 139, 158
  - use
    - Paging Activity area 31
    - Program Isolation Activity area 13
    - PSB/Transaction area 15
    - System Activity area 15
- DRRN translations 91
- DUMPQ 283
- dumps
  - analysis 511
    - Event Collector 511
    - MAINVIEW AutoOPERATOR for IMS 511
  - AO exit 511
  - IMS 511
  - symptoms 511
- DWAIT
  - current period 436
  - description 445
  - event definitions 450
  - MWAIT display 433
  - requesting 443
- DWARN
  - definition 58
  - monitor service title 56
  - warnings 57
- DXRJPROC
  - IRLM 308, 314, 348, 354
  - IRLMG 317
- dynamic transactions 253
- dynamic users 428

## E

- elapsed time 131, 152
  - active region 162
- EMHB pool
  - DPOOL display 265
  - POOLS display 297
- enqueue
  - held 140, 159, 285
  - messages to LTERM 431
  - outstanding 284
- EPCB pool
  - DPOOL display 268
  - POOLS display 297
  - tuning 27
- error buffers
  - OSAM 217
  - VSAM 231
- ESDS inserts 230
- ESDS resource ID 286
- ETO
  - dynamic terminals 395

- Event Collector
  - DREGN display
    - with data from 125
    - without data from 145
  - dump analysis 511
- events
  - application program 451
  - DWAIT definitions 450
  - DWAIT queue exclusion 450
  - input communications 468
  - input queue 477
  - output queue 481
  - scheduling 484
  - synchronization point 498
- exclusive
  - enqueue 159
  - lock
    - current state 321, 330, 362, 372
    - held 328
    - waiters 327, 336, 369, 379
  - mode, users 428
- exclusive locks 337
- exit routines
  - IRLM 310, 317, 351, 357
- EXPAND feature
  - APPC 237
  - APPCL 244
  - DAPPC 250, 255
  - LATCH 273
  - LCUSR 328, 370
  - LHUSR 337, 380
  - LUSRD 339, 383
  - PI 284
  - POOLS 293
  - USER 430
  - VSST 226
- external storage
  - VSAM access 231

## F

- Fast Path 223
  - BALG queuing 9
  - buffer
    - overflow 286
    - pool 223
    - size 224
    - wait 466
  - checkpoint synchronization 276
  - command 276
  - DEDB resource ID
    - PI service 286
  - DEDB share area name
    - DLTCH service 258
  - DEDB sync area name
    - DLTCH service 259
  - EPCB pool 270, 297
  - exclusive enqueue 159
  - Expedited Message Handler 240, 252

- Fast Path (continued)
  - FPWP pool 265, 297
  - IFP
    - DPOOL 265
    - POOLS 297
  - lock 322, 323, 331, 332, 343, 345, 363, 364, 373, 374, 389, 390
  - MSDB name
    - DLTCH service 259
  - MSDB resource ID
    - PI service 286
  - resource latch 461
  - SYNC point latch 500
  - transaction waiting 478
- Fast Path buffer pool
  - synchronization point latch 500
- FETCH
  - I/O active 488
  - I/O queued 489
- fetch request elements (FREs) 84
- filters
  - workload wait 437
- fixed pools
  - APPC 265, 297
  - block expansions/compressions 264, 267, 295
  - buffers 267
    - tuning tips 267, 292, 298
  - net expansions 264, 267, 295
  - non-CBT 293
  - ownership 265
  - size 264
  - wasted space 266, 298
- flow components 484
  - application program 451
  - input communications 468
  - input queue 477
  - output communications 480
  - output queue 481
  - synchronization point 498
- FP
  - SET operand 439
- FPU (SET operand) 438
- FPWP pool
  - DPOOL display 265
  - POOLS display 297
- fragmented pool 271
  - MFS 88
- FRE
  - MFS queue 86
  - unavailable 472
- FRE (fetch request element) 84, 86
- free block request 85
- free space
  - MFS pool 87, 88
  - OSAM buffer pool 222
  - program isolation pool 285
- FREEMAIN 289
- frequency distribution 53
- FSEL(+) field 237

## G

- generic latches
  - contention statistics 272
- generic qualifier 75
- get next requests, OSAM 216
- GET NEXT within PARENT calls 134
- GETMAIN 289
- global
  - deadlocks 318, 358
  - IRLM display
    - sharing group 350, 357
  - lock 322, 331, 344, 363, 373, 389
- group
  - data sharing
    - IRLM 347, 350, 354, 357
- grouping requests 62

## H

- HDAM
  - lock 322, 331, 344, 364, 374, 389
  - resource ID 286
- held enqueues 140, 159
- held resource 322, 331, 343, 363, 373, 388
- high-water mark 53
  - CBT pool 289
  - non-CBT pool percentage
    - DPOOL display 264, 269
    - POOLS display 295
  - non-CBT pools
    - allocation size 264, 269, 296
  - sequential buffer storage 218
- HIOP pool
  - DPOOL display 265
  - POOLS display 297
  - tuning 27
- Hiperspace
  - page stolen 232
  - reads 232
  - writes 232
- Hiperspace buffers 18
  - number defined 228
  - statistics 232
  - subpool number 230
  - VSAM control interval 228, 231
- historical data
  - long-term 53
  - short-term 53
- history
  - DWAIT display 447
- HISTORY (SET keyword)
  - use 441
- hit ratio 17
  - analyzer displays
    - DBST OSAM global 217
    - DBST OSAM subpool 221
    - VSST VSAM global 227
    - VSST VSAM subpool 228, 231
  - VSAM 232

- HOT (SET operand)
  - restart
    - MWAIT service 437
- HPHIT monitor 18
- hyperlink
  - DREGN 147

## I

- I/O
  - application program 451
  - errors 86
  - paged-fixed blocks 230
  - queuing 92
  - request processing 85
  - service units 142, 162
  - VSAM 231
  - WADS requests 282
- IC
  - DWAIT input communications events 468
- identify request 309, 315
- IFP
  - DPOOL 265
  - GU 478
  - POOLS 297
- IFP (SET operand) 438
- Image log 61
- immediate request 86
- IMRPRINT 60
- IMS
  - dump analysis 511
  - internal services displays 233
  - Manager 32
  - monitoring 3
  - restart
    - global pool statistics 215
    - subsystem ID 339, 384
- IMS performance analyst and system programmer 33
- IMS RESOURCE ANALYZER
  - services 63
  - techniques 1
- IMS RESOURCE MONITOR
  - techniques 1
- IMS WORKLOAD ANALYZER
  - services 63
  - techniques 1
- IMS WORKLOAD MONITOR
  - techniques 1
- IMSPLEX
  - description 348, 355
- IMS-related data 252
- in-core format index 85
- index buffer 230
- information 12
- input
  - allocation direction 237
  - display 249
  - communications
    - DWAIT IC events 468

- input (continued)
  - conversations 239, 246
  - queue
    - DWAIT IQ events 477
- input queue (IQ) 450, 477
- inserts
  - ESDS 230
  - KSDS 230
- intent failures, TRANSACT macro 486
- intent lists
  - I/O active 492
  - I/O queued 492
- internal functions
  - APPCA 233
  - APPCL 242
  - DAPPC 249, 254
  - LATCH 273
  - logging 279
  - program isolation 284
- interprocessor environment
  - IRLM 308, 314, 348, 354
- interval
  - historical data 53
  - specifying 436
- INTERVAL (SET keyword) 436
  - use 441
- intraprocessor environment
  - IRLM 308, 314, 348, 354
- IQ
  - DWAIT input queue events 477
- IRLM
  - data sharing group 347, 354
  - identified subsystems 319
  - intraprocessor/interprocessor environment 308, 314, 348, 354
  - lock types 323, 344, 374, 390
    - 332, 364
  - partner
    - failure 310, 316
    - lock holders 322, 331, 345, 363, 373, 388
    - lock waiters 324, 333, 346, 365, 391
  - recovery 308, 314
  - release number 308, 314, 348, 354
  - request contention 348, 355
  - scope 308, 314, 348, 354
  - services
    - IRLM (IRLM 2.1 and later status) 347
  - sharing subsystems 308, 314, 348, 354
  - status 350, 357
  - status exit 309, 315, 350, 356
  - subsystem name 308, 314, 348, 354
  - suspensions 349
  - VTAM APPL names 317
  - wait 461
    - sync point 503
- XES
  - lock table 347, 354
  - OSAM table 348, 354
  - VSAM table 348, 354

- IRLM services
  - IRLM (IRLM 1.5 status) 307
  - IRLMG (IRLM 1.5 global status) 313
  - IRLMG (IRLM 2.1 and later global status) 353
  - LCRES (lock contention by resource) 321, 362
  - LCUSR (lock contention by user) 328, 370
  - LHRES (locks held by resources) 330, 372
  - LHUSR (locks held by user) 337, 380
  - LUSRD (lock user detail) 339, 383
- ISI security 424
- ISL 311, 352
- ISTAT 395
- ITASK 468, 471, 474
  - analysis 27
  - DWAIT
    - IC events 468, 469, 474
    - SC events 491

## J

- Journal 35

## K

- key retrieves 216, 231
- KSDS
  - inserts 230
  - resource ID 286

## L

- LATCH 272
  - ACTL 277
  - APSB 274
  - BLKM 277
  - CBTS 274, 452, 485, 498
  - CCTL 275
  - contention 258
  - CONV 275
  - DBAU 275
  - DBBP 276
  - DBLK 275
  - DBLR 274
  - DBSL 275
  - DC Monitor 487
  - DCSL 275
  - DDRB 275
  - DDRP 275
  - DEDB 499
  - DISP 275
  - DMBB 275
  - DMBP 275
  - DMSH 275
  - DMSY 275
  - EXPAND 273
  - Fast Path 276
  - Fast Path SYNC point 500
  - generic 489
  - logger 493

- LATCH (continued)
  - LOGL 276
  - LUBT 274
  - LUML 276
  - MSDB 276, 505
    - transactions waiting 463
  - other 496
  - PDRB 276
  - PDRP 276
  - PI enqueue/dequeue 465, 496
  - PSBB 276
  - PSBP 276
  - QBSL 276
  - QMGR 276
  - SCHD 277
  - SMGT 277
  - storage management 497
  - SUBQ 277
  - TCTB 277
  - TERM 277
  - USER 277
  - VLQB 274
  - VTCTB 274
  - wait time 278
  - XCNQ 276
- latch
  - CBTS 468
  - DC Monitor 455, 469
  - generic 461, 469
  - logical log 470
  - other 464, 473
  - overflow buffer 465
  - PI enqueue/dequeue 473
  - storage management 466
- latch summary 273–278
- LCRE 282, 283
- LCRES (lock contention by resource) display 321, 362
- LCUSR (lock contention by user) display 328, 370
- LGMSG 24
  - I/O active 469, 493
    - application program 462
    - sync point manager 503
  - I/O queued 470, 493
    - application program 462
    - sync point manager 503
- LHRES (locks held by resources) display 330, 372
- LHUSR (locks held by user) display 337, 380
- lines
  - busy 481
  - unavailable 481
- local lock 322, 331, 344, 363, 373, 389
- locate requests
  - OSAM subpool 220
- locates 216
- lock
  - deadlocks 358
  - global resource request 349, 356
  - local resource request 349, 356
  - request 309, 315
- lock manager
  - local 308, 314, 348, 354
- lock table
  - XES, IRLM 347, 354
- locks
  - analyzer
    - DREGN program isolation activity 139, 158
    - PI service 284
  - contention
    - dependent regions 328, 370
  - deadlocks 318
  - Fast Path 323, 332, 345, 364, 374, 390
  - held 126
  - held by region 328, 346, 391
  - holder 284
  - holders 363, 373
  - in transition
    - LCRES (IRLM 1.5) 321, 327
    - LCRES (IRLM 2.1) 362, 369
    - LHRES (IRLM 1.5) 330, 336
    - LHRES (IRLM 2.1) 372, 379
- PTB
  - process 312, 319
  - requests 308, 314
- regions waiting for 338
  - LCUSR (IRLM 1.5) 329
  - LCUSR (IRLM 2.1) 371
  - LHUSR (IRLM 1.5) 338
  - LHUSR (IRLM 2.1) 381
  - LUSRD (IRLM 1.5) 343
  - LUSRD (IRLM 2.1) 388
- request handler requests 308, 315
- state 345, 346, 390, 391
- structure capacity 359
- table 360
- types 323, 332, 344, 364, 374, 390

- log buffer wait 470
- LOG DISPLAY 55
- log statistics (LOGST) 279
- logger
- activity 280
- asynchronous write requests 281
- check write requests 281
- latch 493
- wait write requests 282
- logging
- buffer wait 493
- displays 60, 61
- logical logger latch 470
- LOGL latch 276
- logs
- configuration display 279
- long message queue data set 90
- long term
- history 53
- page fixing option 220

- LPA
  - fault 470
  - application program 451
  - control region 489
  - DB2 455
  - dependent region 490
  - DL/I 457
  - SAS 492
  - sync point manager 503
- GFA 471
- LTERM
  - assigned 431
  - input status messages 401
  - messages enqueued 431
  - monitoring 8
  - unavailable 482
- LU
  - allocation directions 246
  - partner 250
  - status 237
- LU name
  - APPCA
    - partner LU 240
    - route connections 239
  - APPCL
    - partner LU 247
    - route connections 246
  - DAPPC (input allocation)
    - partner LU 250
  - DAPPC (output allocation)
    - partner LU 255
  - deallocation requests 256
  - requests queued 241, 247, 256
  - stopped 241, 247, 251, 255
  - trace 247, 251, 255
- LUBT latch 274
- LUMC pool
  - DPOOL display 265
  - POOLS display 297
- LUML latch 276
- LUMP pool
  - DPOOL display 265
  - POOLS display 297
- LUSRD (lock user detail) display 339, 383
- LWA wait 462, 471, 493

## M

- MAIN pool
  - DPOOL display 268, 270
  - POOLS display 297
- main storage service units 142, 162
- MAINVIEW AutoOPERATOR for IMS
  - dump analysis 511
- mapped conversation 251, 257
- mark buffer altered requests 216
- Master Terminal Operator 30
- MDP (SET operand) 438
- measurements 54

- message
  - switches 129, 149
- message formatting service (MFS)
  - MFSUT 87
- messages
  - application program 451
  - enqueued to LTERM 431
  - input communications 468
  - input queue 253, 477
  - output queue 481
  - scheduling 484
  - segment
    - relocate request 91
  - synchronization point 498
  - transaction queue status 423
- MFBP pool
  - DPOOL display 268, 270
  - POOLS display 297
  - tuning 26
- MFS
  - FRE unavailable 472
  - I/O active 471
  - I/O queued 471
  - pool failure 472
  - test mode
    - users 428
- MFS (Message Format Service)
  - block 84
  - block space 88
  - data set
    - first extent 84
  - format block 85
  - in-core format index 85
  - pool 83
    - status 87
    - tuning 11
    - utilization (MFSUT) 87
  - queue 86
  - request statistics 85
- MFSTEST
  - control blocks 26
- MFSUT 87
- mode
  - name
    - conversation 252, 257
- monitor
  - parameter 436
  - request title 56
  - restart 52
    - MWAIT 437
  - service title 56
  - starting and stopping 52
  - stop time 437
  - workload wait selection criteria 437
- monitoring
  - specific resources 439
  - techniques 3

## MPP

- block loader busy 485
- class not active 478
- database intent 486
- DBWP pool 487
- GU 478
- not scheduled 478

MPP (SET operand) 438

MPR busy 477

## MSC

- delays 481
- logical link unavailable 482
- physical link busy 482
- physical link unavailable 482

MSCCLOCK operand (TERM SET keyword) 439

## MSDB

- latch 276, 463, 505
  - transactions waiting 463
- lock 322, 331, 344, 363, 373, 389
- resource ID 286

multiple requests 62

MVIMS sync point processing 499

## MWAIT

- current period 436
- start time 437

## N

name qualifier

- workload analyzer request 75

net expansions

- fixed pools 264, 295

network stopped

- MFS pool 472
- MFSP pool 472
- SAP 474, 475

NOBK indicator 13

NOCOMM 438

node

- busy 482
- ID 429
- name
  - user 431
- unavailable 483
- users allocated 428

non-CBT pools 24–27

- fixed detail 262
- summary 291

normal buffer allocation 136, 157

notify exit 311, 317, 351, 357

notify request

- XCF 350, 357

N-way sharing group 371, 381, 388

## O

OLDS (online log data set) 21, 22

- allocated 280
- archiving 281
- block size 279
- blocks read 281
- blocks written 281
- statistics display 279

operating system

- information 140, 160

## OQ

- DWAIT output queue events 481

OS wait 464, 473

OS390 subpool 289

## OSAM

- buffer pool
  - DBST global pool display 215
  - performance 217
- buffers 217, 221
  - latch 276
- HDAM resource ID
  - PI service 286
- I/O active 464, 505
- I/O queued 464, 505
- pools (DBST) 215
- queuing 6
- reads 217
- resource ID 286
- segment contention 286
- writes 16, 217

OSAM buffer pool 21

OSAM table

XES

- IRLM 348, 354

output

- allocation direction
  - display 254
- communications
  - DWAIT events 480
- conversations 239, 246
- delays 481
- messages 481
- queue
  - DWAIT OQ events 481
- threads 224

output queue 450

overflow buffer

- allocation 225
- latch 465

overflow buffer allocation 136, 157

## P

page fault 502

- application program 452

CSA GFA

- application program 451

GFA

- application program 452

- page fault (continued)
  - GFA queue 471, 474
  - LPA 470
  - message processing 469
  - private area 473, 491
  - sync point 502
- page fixing option
  - long term 220
- page-fixed buffers 230
- paging
  - activity 148
- parameter
  - resource monitor request 436
- partner IRLM
  - failure 310, 316
  - lock contention 330, 372
  - lock holders
    - LCRES (IRLM 1.5) 322
    - LCRES (IRLM 2.1) 363
    - LHRES (IRLM 1.5) 331
    - LHRES (IRLM 2.1) 373
    - LUSRD (IRLM 1.5) 345
    - LUSRD (IRLM 2.1) 388
  - lock waiters
    - LCRES (IRLM 1.5) 324
    - LCRES (IRLM 2.1) 365
    - LHRES (IRLM 1.5) 333
    - LHRES (IRLM 2.1) 375
    - LUSRD (IRLM 1.5) 346
- partner LU 240, 247, 250, 255
- PCB 92, 123
- PDRB latch 276
- PDRP latch 276
- percent data measurement 54
- period 60
- PI
  - enqueue/dequeue latch 465, 496, 506
- PI (program isolation) 9, 13
  - activity 128, 148
  - display service 284
  - enqueue/dequeue latch 473
- PIENQ
  - example of 35
- plan name 137
- P-lock 322, 331, 344, 363, 373, 389
- PLOT 53
  - definition 58
  - range distribution 58
- plot 30–35
- pool analysis 10–12
- pool space failure 10, 11
- POOLC 288
- POOLS
  - EXPAND 293
- pools 291
  - current definition 216
  - fixed 293
  - FPBST, Fast Path buffer pool 223
  - index buffers 230
- pools (continued)
  - MFS 83, 87
    - failure 472
    - fragmentation 88
    - free space 88
  - MFSUT, MFS pool 87
  - ownership 270, 297
  - program isolation 11, 284
  - program isolation (PI) 284
  - tuning tips 295
  - variable 293
  - VSST, VSAM pool 226
  - VSST, VSAM subpool 229
- pop-up window 333, 365, 375
- positional qualifier 75
- positionally 75
- PREFETCH 86
- preset mode
  - users 429
- priority cutoffs 9
- PROCLIM
  - reschedule 479
- product support iii
- program isolation 473
  - activity 128, 148
  - DREGN 139, 158
  - pool tuning 11
  - service 284
- PSB
  - I/O active 496
  - I/O queued 496
  - work area 138, 158
- PSB name
  - DREGN 129, 149
  - LCUSR 328, 370
  - LHUSR 337, 380
  - LUSRD 340, 384
- PSB pool
  - DLMP 26, 270, 297
  - DPSB 26, 270, 297
  - intent failure 9
  - LSO 11
  - resident 26
  - tuning 11
  - use 10, 26
- PSBB latch 276
- PSBC pool
  - DPOOL display 270
  - POOLS display 297
- PSBD pool
  - DPOOL display 270
  - POOLS display 297
- PSBNAME 253
- PSBP latch 276
- PSBW pool
  - description 26
  - DPOOL display 268, 270
  - POOLS display 297
  - use 26

PST 340, 384  
 PTB  
   lock requests 308, 312, 314, 319  
   size 318  
 PUR (SET operand)  
   restart  
     MWAIT service 437  
 purge calls 227  
 purge request  
   IRLM 309, 315, 350, 356  
   OSAM subpool 220  
 purging  
   requests  
     MWAIT restart 437  
     QUEST 91  
   wait 93

## Q

QBLKS  
   data set 90  
   I/O active 465, 496  
   I/O queued 465, 497  
 QBSL latch 276  
 QBUF pool  
   DPOOL display 268, 270  
   POOLS display 297  
   tuning 24  
 QIS  
   SET keyword  
     MWAIT service 436  
   SET operand  
     MWAIT service restart 437  
 QMGR latch 276  
 qualifier 75  
 query request 309, 315, 350, 356  
 QUEST service 89  
 queue  
   buffer size 470  
   data set  
     block size 89  
     device type 90  
     record length 90  
     volume 90  
   dead letter 428  
   events  
     excluding from DWAIT display 450  
   I/O activity 92  
   long message data set 90  
   manager  
     calls 91  
     requests 91  
   statistics (QUEST) 89  
   type 90  
   users 429  
   waits 92  
 queued transactions, resource status 89  
 queuing analysis 6–7  
 quit request 309, 315

## R

range distribution 58  
 RBA 286  
   lock display 322, 331, 344, 363, 373, 389  
 RBA retrieves 231  
 read only lock  
   LCRES (IRLM 1.5) 321, 327  
   LCRES (IRLM 2.1) 362, 369  
   LHRES (IRLM 1.5) 330, 336  
   LHRES (IRLM 2.1) 372, 379  
 reads  
   hiperspace 232  
   OSAM 217, 221  
 real storage analysis 28  
 RECA pool 27  
 record length  
   queue data sets 90  
 record lock 322, 331, 344, 363, 373, 389  
 recovery  
   IRLM 308, 314  
 recovery block 282  
 region  
   execution 141, 160  
   holding resource 287  
   identification  
     DREGN service 129, 149  
   status  
     DLIST service 119  
     DREGN service 129, 149  
   type 131, 152  
 REGION (SET keyword) 438  
 region displays  
   detail data 125, 145  
 REGND  
   program isolation 139, 158  
 REGNS  
   use 14  
 request  
   contention  
     IRLM 348, 355  
   IRLM  
     suspended 349  
   multiple 62  
   status 62  
 request handler lock requests 308, 315  
 requests  
   free block 85  
   message segment  
     relocate 91  
   notify  
     XCF 350, 357  
   prefetch 85  
   queue manager 91  
 Resource Analyzer  
   description 49  
 resource analyzer  
   services 63  
 resource displays 215  
 resource ID 286

- Resource Monitor 49
- resource selection 439
- resource status
  - queued transactions 89
- resources
  - IRLM request contention 348, 355
- response mode
  - users 429
- response time
  - DWAIT display 447
  - workload wait events 433, 445
- restart
  - global pool statistics 215
- restart service
  - MWAIT 437
- resume exit 311, 317, 351, 358
- retrieves
  - key 216, 231
  - RBA 231
  - VSAM 231
- rgdependent regions
  - activities and resources 41
  - services 41
- RID 322, 331, 344, 363, 373, 389
- RPL blocks
  - storage 227
- RST (SET keyword)
  - MWAIT service 437

## S

- sampling interval
  - MWAIT 441
  - specifying 436
- SAP (save area prefix) 28
  - shortage 475
  - unavailable 474
- SAP pool
  - tuning 24
- SAS
  - fault 492
  - I/O active
    - SYNC point 506
  - I/O queued
    - SYNC point 506
  - SYNC I/O active 497
  - sync I/O active 466
  - SYNC I/O queued 497
  - task wait 467
- SBBPROF/BBPROF considerations 513
- SC
  - DWAIT scheduling events 484
- SCHD latch 277
- scheduling
  - DWAIT SC events 484
  - priority 138, 157
- scope
  - IRLM 308, 314, 348, 354
- scrolling
  - APPCA 238
  - APPCL 245
  - DAPPC 250, 255
  - ISTAT 396
  - LATCH 274
  - PI 284
  - POOLS 294
  - USER 431
  - VSST 226, 229
- search buffer requests
  - VSAM 231
- security
  - ISI 424
- segment 138, 158
- segment contention
  - OSAM block 286
- selecting
  - IMS subsystem activities 41
- selection criteria
  - workload wait 437
- sequential buffering 129, 149, 218
- service select code 284
  - APPCA 233
  - APPCL 242
  - DAPPC 249, 254
  - DLTCH 258
  - DPOOL 262, 268
  - DWAIT 445
  - FPBST 223
  - IRLM
    - IRLM 1.5 307
    - IRLM 2.1 347
  - IRLMG
    - IRLM 1.5 313
    - IRLM 2.1 353
  - ISTAT 396
  - LATCH 273
  - LCRES 321, 362
  - LCUSR 328, 370
  - LHRES 330, 372
  - LHUSR 337, 380
  - LOGST 279
  - LUSRD 339, 383
  - MFSST 83
  - POOLC 288
  - POOLS 292
  - QUEST 89
  - USER 427
  - VSST 226
- service units 140, 142, 160, 162
- service(s)
  - analyzers 63
  - database 215
  - IMS internals 233
  - IRLM
    - 1.5 307
    - 2.1 and later 347

- service(s) (continued)
  - IRLMG
    - IRLM 1.5 313
    - IRLM 2.1 and later 353
  - MFS 83
  - region activity
    - DREGN 125, 145
  - title 55, 56
  - user status 427
  - workload wait 445
- SET request 53, 61
  - block requests 62
- SEV1 status 423
- SEV2 status 423
- SEV3 status 423
- share lock
  - held 328, 337
  - LCRES (IRLM 1.5) 321, 327
  - LCRES (IRLM 2.1) 362, 369
  - LHRES (IRLM 1.5) 330, 336
  - LHRES (IRLM 2.1) 372, 379
- sharing group
  - IRLM 348, 355
  - lock structure capacity 359
  - N-way 371, 381, 388
- sharing subsystems
  - IRLM 308, 314, 348, 354
- SHMSG 24
  - I/O active 466, 475, 497
  - I/O queued 466, 476, 497
- short message queue
  - data set 90
- shutdown 52
- SIDEINFO 240, 247, 255, 257
- size
  - fixed pools 264
    - allocation 264
    - percentage 264
  - OSAM subpool 219
  - pools 269, 296
    - percentage 295
    - POOLS display 295
  - subpool 230
  - variable pools
    - allocation 269
    - percentage 269
- SLDS (system log data set) 22
- SLM 349, 350, 355, 356
- SMGT latch 277
- snap request 309, 315
- SNAPQ 283
- SORT
  - APPCA 235
  - APPCL 244
  - DPOOL 262
  - POOLS 292, 293
  - USER 429
- SP
  - DWAIT synchronization point events 498
- SPA
  - I/O active 476
  - I/O queued 476
- space
  - wasted
    - fixed pools 266, 298
- space release algorithm 87
- SPQB 432
  - blocks 277
- SQL call
  - DAPPC 253
  - DWAIT event 453
- SSQ status 423
- START (SET keyword)
  - MWAIT service 437
- start time
  - MWAIT request 437
- starting
  - monitors 52
  - workload wait 433
- startup 52
- STAT call 124
- STAT/STATR
  - techniques 12
- static users 427, 428, 430
- statistics
  - pool
    - global 215
- status
  - APPC connection 238, 245
  - data measurement 54
  - exit
    - routine call 311, 317, 351, 358
    - subsystem response 309, 350, 356
  - IRLM 350, 357
  - MFS pool 87
  - requests 62
  - resource 89
  - user 431
- status exit
  - IRLM 315
- STM task compression 289
- STOP (SET keyword)
  - MWAIT service 437
- stop time
  - monitor request 437
- stopped
  - users 430
- stopping
  - monitors 52
  - workload wait 440
- storage
  - external access 228
  - management latch 466, 476
  - pool allocation
    - fixed pool 264
    - POOLS display 294, 296
    - variable pool 269
  - RPL blocks 227

- storage (continued)
  - sequential buffering maximum 218
  - service units 142, 162
  - VSAM pool 227
- storage management
  - latch 497
- structure capacity
  - locks 359
- subpool
  - ID 228
  - OSAM
    - current definition 219
    - database 216
    - display 219
    - number of requests 220
    - performance 220
    - size 219
  - relative number 228
  - requests 230
  - size 228, 230
  - type 228
- SUBQ latch 277
- subsystem
  - identified to IRLM 319
- subsystem ID 339, 384
  - IRLM 308, 314, 348, 354
- summary periods 53
- support, customer iii
- suspend exit 311, 317, 351, 358
- suspends 312, 319
  - IRLM request 349
- swapping status 143, 162
- SYNC 498
- synchronization
  - level 252, 257
- synchronization point (SP) 498
- synchronous conversations 239, 246
- synchronous notify request 309, 315, 350, 356
- SYNCLOCK operand (TERM SET keyword) 439
- syntax conventions xvii
- system activity 128, 148

## T

- takeover request 309, 316, 350, 356
- TCB
  - CPU time 143, 162
  - execution status 141, 160
- TCTB latch 277
- technical support iii
- TERM (SET keyword) 439
- TERM latch 277
- terminals
  - dynamic 395
- test mode
  - users 430
- thread
  - output 224

- threshold
  - defining 55
  - exceeding 56
  - use 52
  - value 57
  - warning message 57
- time 11
- time interval 436
  - specifying
    - MWAIT request 436
- timer request panel
  - workload wait request 435
- title
  - defining of 56
- token 124
- TP
  - requests queued 247, 256
- TP name
  - confirmations 256
  - conversation initiation 241, 251, 257
  - deallocation requests 256
  - message type received 240, 252
  - requests queued 247, 256
  - stopped 241, 248, 256
  - trace 256
  - transaction code scheduled 241, 253
- TPI
  - DAPPC (input allocation) 253
- TPI (SET operand)BMP (SET operand) 438
- trace
  - DL/I buffer handler
    - DBST 216
    - VSST 227
  - LU name 247, 251, 255
  - TP name 256
- TRANQ 250
- TRANSACT macro 240, 241, 252
- transaction flow
  - component display 445
  - DWAIT events 450
- transactions
  - APPC
    - type 253
  - conversational 138, 157, 240
  - dynamic 253
  - elapsed time 131, 152
  - response time
    - display 447
  - scheduling priority 138, 157
- translations
  - DRRN 91
- TRANTYPE (SET keyword) 439
- tuning tip
  - LOGST 282

## U

- UBBPARM/BBPARG considerations 513
- UBBSAMP/BBSAMP considerations 513
- unlock
  - request 309, 316, 350, 356
- update lock 327
  - held 328, 337
  - LCRES (IRLM 1.5) 321
  - LCRES (IRLM 2.1) 362, 369
  - LHRES (IRLM 1.5) 330, 336
  - LHRES (IRLM 2.1) 372, 379
- USER
  - EXPAND 430
  - latch 277
- user
  - active conversation 428
  - dynamic 428
  - exclusive mode 428
  - MFS test mode 428
  - node name 431
  - preset mode 429
  - response mode 429
  - static 430
  - static ISC 428
  - stopped 430
  - test mode 430
- user ID 430
- USERID (SET keyword) 439

## V

- variable pools
  - allocated blocks 271
  - fragmentation 271
  - free spaces 271
  - non-CBT 293
  - size 269
- verify request 309, 316, 350, 356
- VHIT monitor 17
- VIO slots 143, 163
- VLQB latch 274
- volume
  - queue data set 90
- VSAM
  - buffer handler
    - requests 230
  - buffer pool 226
  - control interval 228
  - database pool 227
  - error buffers 231
  - ESDS resource ID
    - PI service 286
  - external storage 231
  - get requests 231
  - hiperspace
    - buffers 228
    - reads 232
    - writes 232

- VSAM (continued)
  - I/O 231, 466
    - active 506
    - queued 507
  - KSDS resource ID
    - PI service 286
  - lock 323, 332, 344, 364, 374, 389
  - retrieve requests 231
  - search buffer requests 231
  - subpool 228
    - defined 230
    - ID 228, 230
    - performance 227, 228
  - writes
    - user 228, 232
- VSAM table
  - XES
    - IRLM 348, 354
- VSBP
  - latch 469
  - lock 461, 489
- VSST 17, 18
  - service description 226
- VTAM 472
  - MFS 472
  - node
    - busy 482, 483
    - save area prefix (SAP) 474, 475
- VTCL latch 274

## W

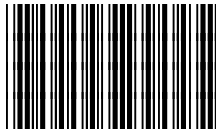
- WADS (write-ahead data set) 22
  - block size 279
  - buffer 281
  - I/O requests 282
  - segments written 282
  - statistics display 279
- wait
  - buffer enqueue/dequeue 92
  - buffer handler 221
  - DECB 93
  - IMS internal 473
  - latch 274
  - OS 473
  - PI 284
  - purge 93
  - queuing 92
  - resource 322, 331, 343, 363, 373, 388
  - workload 435
- wait time
  - latch 278
- wait write requests 282
- warning condition
  - description 55
- warning message 56
  - format 56
  - ID 56
  - number of 55

- warning message (continued)
  - sending of 55
  - time interval 55
  - title 56
- wasted space
  - fixed pools 266, 298
- WIF (SET keyword) 55
- WIN (SET keyword) 55
- WKAP pool
  - DPOOL display 270
  - POOLS display 297
  - use 24
- WLIM (SET keyword) 55
- WMAX keyword 55, 57
- WMSG (SET keyword) 55
- work area
  - PSB 138, 158
- workload
  - events 59, 445
    - definitions 450
  - trace 59
  - wait
    - selection criteria 437
  - wait data entry panel 435
- Workload Analyzer
  - displays, alphabetical reference 393
  - DWAIT display 445
- workload analyzer
  - services 63
- Workload Monitor
  - description 49
- Workload Wait (DWAIT) 443
- writes
  - background 227, 232
  - hiperspace 232
  - OSAM 217, 221
  - VSAM 228, 232
- written blocks
  - OSAM subpool 220
  - total OSAM 217
- WTO (write-to-operator SET operand) 55
- WVAL keyword
  - threshold, warning message 55, 57

## X

- XCF
  - notify request 350, 357
- XCNQ latch 276
- XES 347, 348, 354
  - delay 359
- XRF
  - backup request 309, 316, 350, 356
  - IRLM 310, 316
  - LU name 239, 246
  - takeover 323, 332, 345, 364, 374, 390

## Notes



\*22782\*